

# Privacy-Friendly Photo Sharing and Relevant Applications Beyond

THÈSE N° 7828 (2017)

PRÉSENTÉE LE 28 JUILLET 2017

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

GROUPE EBRAHIMI

PROGRAMME DOCTORAL EN GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Lin YUAN

acceptée sur proposition du jury:

Dr J.-M. Vesin, président du jury  
Prof. T. Ebrahimi, directeur de thèse  
Prof. S. Voloshynovskiy, rapporteur  
Prof. P. Schelkens, rapporteur  
Prof. J.-Ph. Thiran, rapporteur



ÉCOLE POLYTECHNIQUE  
FÉDÉRALE DE LAUSANNE

Suisse  
2017



*Look deep into nature,  
and then you will understand everything better.*  
— Albert Einstein

To my parents and family...





# Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Prof. Touradj Ebrahimi for the continuous support of my PhD study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Jean-Philippe Thiran, Prof. Peter Schelkens, Prof. Slava Voloshynovskiy, and Dr. Jean-Marc Vesin, for the time they have taken to review my manuscript, and for their valuable comments.

I would like to thank all my current and former colleagues from the Multimedia Signal Processing Group: Martin, David, Anne-Flore, Irene, Evgeniy, Evangelos, Ashutosh, Christine, Philippe, Pavel, Ivan, Grégoire, Margherita, He and Hiromi. We always share ideas as well as stories during our lunch time, which kept me motivated and enthusiastic. I would like to thank the students I have worked with, Joël, Thierry, Thomas, Vincent and Pablo, for helping me in development and implementation of important softwares used in this thesis. A special thank goes to Dr. David McNally for providing supports in IT infrastructures and helpful discussions.

My sincere thanks also go to all my Chinese friends for providing me with a lot of mental supports, knowledges and happiness: Yuanhao Bi, Yizhong Liang, Yan Zhou, Xin Tang, Qiang Wang, Xin Lu, Su Li, Lin Bai, Yuliang Zheng, Junrui Zhang, Zhisheng Yang, Ming Li, Xiaoyan Wang, Tao Guo, Feng Zhang, Hong Jian, Huan Liu, Guanchun Li, Shiqi Wang, Tao Yu, Guobi Zhao, Yu Yu, Xiaojiang Peng and Yang Wang.

Last but not the least, I would like to thank my family: my parents, grandfather, aunts, uncles, sisters and especially my girlfriend Xiao, for supporting me spiritually throughout writing this thesis and my life in general.

Lausanne, 21 June 2017

Lin Yuan



# Abstract

Online photo sharing has become everyday life for many, but has also raised concerns for privacy. Online social networking (OSN) sites usually offer a limited degree of privacy protection and the most common solution is conditional access control. Researchers have studied various approaches to enable image privacy in photo sharing, mostly focusing on encrypting or distorting image visual content, which may compromise utility and user experience of photo sharing. This thesis investigates novel solutions to protect image privacy with a particular emphasis on the scenario of online photo sharing. To this end, we investigate not only algorithms to protect visual content in image but also design of architectures for privacy-preserving photo sharing. Beyond privacy, we also explore the additional impacts and potentials of employing daily images being captured and shared for other relevant applications.

First, we propose and study two image encoding algorithms to protect image visual privacy, within a framework named *Secure JPEG*. The first method scrambles a JPEG image by randomly changing the signs of its quantized DCT coefficients based on a secret key. The second method, called *JPEG Transmorphing*, allows one to obfuscate arbitrary image regions, while secretly preserving information about the original regions in application segments of the obfuscated JPEG image. Both algorithms are backward compatible with JPEG, meaning that the protected image is readable by any JPEG decoder in its visually protected form; The original image can only be recovered by a dedicated JPEG decoder with the right secret key provided. Evaluations (both objective and subjective) reveal a good degree of storage overhead and privacy protection capability using both methods. Particularly, JPEG Transmorphing proved to be able to preserve the maximum pleasantness from both perception and usage perspectives.

Second, we investigate the design of two architectures for privacy-preserving photo sharing. The first architecture, named *ProShare*, aimed to enable secure and efficient access to user-posted images protected by Secure JPEG, is built based on a public key infrastructure (PKI) integrated with a ciphertext-policy attribute-based encryption (CP-ABE). We implemented and demonstrate the correct and efficient functioning of the ProShare architecture based on both iOS and Android mobile platforms. The second architecture is called *ProShare S*, in which a service provider helps users make photo sharing decisions automatically based on their past decisions made in different contexts. Based on machine learning, the photo sharing service analyzes not only the content of a user's photo, but also the context information about the image capture and a prospective

## Acknowledgements

---

requester, and finally decides whether or not to share the particular photo with that requester, and if yes, at which granularity. We validated the ProShare S architecture with a user study of 23 subjects and extensive evaluation analysis.

As the last part of the thesis, we research into three relevant topics in regard to daily images captured or shared by people, but beyond their privacy implications. In the first study, we adopt the idea of JPEG Transmorphing and propose *aJPEG*, an animated image format based on JPEG compression. The aJPEG provides smaller file size and better image quality compared to conventional Graphics Interchange Format (GIF). In the second study, we attempt to understand the impact of popular image manipulations applied in online photo sharing on evoked emotions of photo observers. It reveals that image manipulations indeed influence people's emotion, but such impact highly depends on image content. By learning from image features such as color and texture, we train and evaluate a simple regressor that is able to accurately predict emotions induced by image manipulation. In the last study, we target on the problem of dietary management using daily photos captured by people. To this end, we employ a deep convolutional neural network (CNN), the GoogLeNet model, to perform automatic food image classification and categorization. The promising results provide meaningful insights in design of automatic dietary assessment system based on multimedia techniques, e.g. image analysis.

**Keywords:** privacy, security, social network, photo sharing, JPEG, Secure JPEG, Scrambling, Transmorphing, aJPEG, backward compatibility, public key infrastructure, ciphertext-policy attribute-based encryption, machine learning, context, decision making, emotion, deep learning





# Résumé

La vulgarisation et la facilitation du partage d'images en ligne soulève des préoccupations concernant la confidentialité des informations partagées. Les chercheurs dans les domaines du traitement d'image et de contenu multimédia ont proposé diverses approches pour permettre la protection des images lors du partage de photos. Cependant, la plupart de celles-ci concentrent leurs efforts sur l'encryptions et les distorsions visuelles. Dans cette thèse, nous étudions de nouvelles solutions pour protéger la confidentialité des images dans le contexte du partage de photos en ligne. Pour ce faire, nous proposons non seulement des algorithmes pour protéger la confidentialité visuelle du contenu d'image, mais aussi des modèles d'architecture pour la conservation des données privées. De plus, des applications potentielles et pertinentes, ne relevant pas de la protection de données, furent aussi envisagées.

Tout d'abord, nous proposons et étudions deux algorithmes de compression d'image pour protéger la confidentialité des informations visuelles de l'image, dans le contexte de Secure JPEG. La première méthode bruite une image JPEG en modifiant aléatoirement les signes de ses coefficients DCT en fonction d'une clé secrète. La deuxième méthode, nommée JPEG Transmorphing, permet de protéger une région d'image en lui appliquant n'importe quelle manipulation, tout en préservant secrètement les régions d'image originales dans les segments d'application de l'image JPEG protégée.

Deuxièmement, sur la base des algorithmes de protection de données Secure JPEG, nous étudions la conception de deux architectures pour la conservation des données confidentielles. La première architecture s'appelle ProShare, construite en fonction d'une infrastructure de clé publique (PKI) intégrée à un cryptage basé sur des attributs de polices chiffrées (CP-ABE). Dans ProShare, une photo est protégée par un algorithme de protection Secure JPEG avec une clé secrète. La photo à protéger peut alors être gardée en toute sécurité sur un service non sécurisé (serveur, nuage, etc.). Aussi, la clé secrète est partagée secrètement avec d'autres personnes avec l'aide de la PKI et du CP-ABE. La deuxième architecture s'appelle ProShare S, dans laquelle un fournisseur de services de partage de photos aide les utilisateurs à prendre des décisions concernant le partage de photos automatique en fonction de précédentes décisions. Le service de partage de photos analyse non seulement le contenu de l'image d'un utilisateur, mais aussi les informations contextuelles sur la capture d'image et le potentiel destinataire. En utilisant l'apprentissage supervisé, le système prend la décision de partager ou non une photo particulière d'un utilisateur avec un certain destinataire et, si oui, selon quelle

## Acknowledgements

---

mesure de protection.

Finalement, nous investiguons trois applications pertinentes concernant les images capturées ou partagées par des personnes, mais au-delà du cadre de la protection de données. Dans la première étude, nous nous inspirons du JPEG Transmorphing et proposons un format de fichier JPEG animé, nommé aJPEG. aJPEG préserve les marqueurs APP des images animées dans une image JPEG et réduit la taille du fichier en plus d'assurer une meilleure qualité d'image par rapport au GIF conventionnel. Dans la deuxième étude, nous essayons de comprendre l'impact des manipulations d'images appliquées lors du partage de photos en ligne sur les émotions des destinataires. L'étude révèle que les manipulations d'images influencent effectivement l'émotion des personnes, mais cet impact dépend aussi du contenu de l'image. Nous utilisons un réseau de neurones convolutif (CNN), le modèle GoogLeNet, pour effectuer une détection et une catégorisation automatique d'images alimentaires. Les résultats obtenus lors de la classification des images alimentaires/non-alimentaires et la catégorisation de l'image alimentaire sont prometteurs et fournissent des informations intéressantes concernant la conception d'un système d'évaluation alimentaire automatique basé sur des techniques multimédias, comme l'analyse d'image.

**Keywords :** confidentialité, sécurité, médias sociaux, partage de photos, JPEG, JPEG sécurisé, Scrambling, Transmorphing, aJPEG, compatibilité ascendante, PKI, CP-ABE, apprentissage automatique, contexte, prise de décision, émotion, deep learning



# Contents

Acknowledgements	i
Abstract (English/Français)	iii
List of figures	xi
List of tables	xv
<b>1 Introduction</b>	<b>1</b>
1.1 Thesis Outline . . . . .	3
1.2 Summary of Contributions . . . . .	4
<b>2 State of the Art</b>	<b>5</b>
2.1 Privacy Protection in Image Visual Content . . . . .	5
2.2 Privacy Protection in Online Photo Sharing . . . . .	7
2.2.1 Access Control . . . . .	8
2.2.2 Secure Protection of Image Content . . . . .	9
2.3 Privacy Analysis in Image . . . . .	10
2.4 Context-Dependent Information Sharing . . . . .	11
<b>I The Algorithms</b>	<b>13</b>
<b>3 Overview of JPEG and Secure JPEG</b>	<b>15</b>
3.1 Overview of JPEG . . . . .	15
3.1.1 JPEG Compression . . . . .	16
3.1.2 JPEG File Format and Syntax . . . . .	19
3.2 Secure JPEG Framework . . . . .	21
<b>4 Secure JPEG Scrambling</b>	<b>23</b>
4.1 JPEG Scrambling: The Algorithm . . . . .	23
4.1.1 The Scrambling Protection . . . . .	24
4.1.2 The Descrambling Recovery . . . . .	26
4.2 Performance Evaluation . . . . .	27
4.2.1 Storage Overhead . . . . .	27

vii

## Contents

---

4.2.2	Privacy Protection Capability . . . . .	30
4.3	Conclusion . . . . .	36
<b>5</b>	<b>Secure JPEG Transmorphing</b>	<b>37</b>
5.1	JPEG Transmorphing: The Algorithm . . . . .	38
5.1.1	Transmorphing Protection . . . . .	38
5.1.2	Transmorphing Reconstruction . . . . .	43
5.2	Performance Evaluation . . . . .	44
5.2.1	Storage Overhead . . . . .	45
5.2.2	Reconstruction Quality . . . . .	46
5.2.3	Privacy Protection Capability . . . . .	51
5.2.4	Pleasantness . . . . .	58
5.3	Discussions . . . . .	62
5.4	Conclusion . . . . .	64
<b>II</b>	<b>The Architectures</b>	<b>65</b>
<b>6</b>	<b>ProShare: Privacy-Preserving Photo Sharing based on a PKI</b>	<b>67</b>
6.1	Cryptography Basics . . . . .	68
6.1.1	Public Key Cryptography and Infrastructure . . . . .	68
6.1.2	Attribute-Based Encryption . . . . .	68
6.2	ProShare: The Architecture Design . . . . .	70
6.2.1	Operating Principle . . . . .	70
6.2.2	Operations . . . . .	72
6.3	Prototype and Implementation . . . . .	76
6.3.1	Functionalities . . . . .	77
6.3.2	Evaluation . . . . .	79
6.4	Conclusion . . . . .	80
<b>7</b>	<b>ProShare S: Context-Dependent Privacy-Aware Photo Sharing</b>	<b>81</b>
7.1	ProShare S: The Architecture Design . . . . .	82
7.1.1	Operating Principle . . . . .	82
7.1.2	Feature Definition . . . . .	83
7.1.3	Photo Sharing Decisions . . . . .	85
7.2	User Study and Data Collection . . . . .	85
7.2.1	The Data Collector . . . . .	85
7.2.2	User Study and Dataset Basic Statistics . . . . .	86
7.3	Evaluation and Analysis . . . . .	87
7.3.1	Methodology . . . . .	88
7.3.2	Within-Subject Analysis . . . . .	88
7.3.3	One-Size-Fits-All Model . . . . .	90
7.3.4	Influences of Features on Decision Making . . . . .	91

---

7.4	Discussions . . . . .	93
7.4.1	System Security . . . . .	93
7.4.2	Automatic Feature Extraction . . . . .	94
7.5	Conclusion . . . . .	96
<b>III</b>	<b>Applications Beyond</b>	<b>97</b>
<b>8</b>	<b>Towards an Animated JPEG</b>	<b>99</b>
8.1	Prior Work . . . . .	101
8.1.1	Graphics Interchange Format (GIF) . . . . .	101
8.1.2	Other Animated Image Formats . . . . .	102
8.2	aJPEG Syntax and Structure . . . . .	103
8.2.1	aJPEG Overview . . . . .	103
8.2.2	aJPEG Header . . . . .	104
8.2.3	aJPEG Inserted Frames . . . . .	104
8.2.4	aJPEG Compressed Image Data . . . . .	105
8.3	Codec and Prototype Applications . . . . .	105
8.3.1	aJPEG Codec . . . . .	105
8.3.2	Prototype Applications . . . . .	105
8.4	Performance Evaluation . . . . .	106
8.4.1	Datasets . . . . .	106
8.4.2	Compression Ratio . . . . .	107
8.4.3	Image Quality . . . . .	108
8.5	Conclusion . . . . .	110
<b>9</b>	<b>Understanding Emotional Impact of Image Manipulation</b>	<b>111</b>
9.1	Prior Work . . . . .	112
9.2	Image Dataset and User Study . . . . .	113
9.2.1	Image Collection and Preprocessing . . . . .	113
9.2.2	User Study based on Crowdsourcing . . . . .	114
9.3	Analyzing Emotions induced by Image Manipulation . . . . .	116
9.3.1	Valence-Arousal Score . . . . .	116
9.3.2	Emotion Keywords Distribution . . . . .	118
9.3.3	Influence of Image Content . . . . .	119
9.4	Predicting Emotions induced by Image Manipulation . . . . .	119
9.5	Conclusion . . . . .	122
<b>10</b>	<b>Towards Dietary Management based on Image Analysis</b>	<b>123</b>
10.1	Prior Work . . . . .	124
10.1.1	Food/Non-food Image Classification . . . . .	124
10.1.2	Food Image Recognition . . . . .	125
10.1.3	Convolutional Neural Network . . . . .	126

## Contents

---

10.1.4 Transfer Learning and Fine-Tuning . . . . .	127
10.2 Datasets . . . . .	128
10.3 Experiments and Analysis . . . . .	130
10.3.1 Food/Non-food Classification . . . . .	130
10.3.2 Food Image Categorization . . . . .	132
10.4 Discussions . . . . .	134
10.5 Conclusion . . . . .	136
<b>11 Summary and Future Work</b>	<b>137</b>
11.1 Thesis Summary . . . . .	137
11.2 Future Directions . . . . .	139
<b>A Screenshots and Supplied Images</b>	<b>141</b>
<b>Bibliography</b>	<b>160</b>
<b>Curriculum Vitae</b>	<b>161</b>

# List of Figures

2.1	Examples of different visual privacy protection approaches: image pixelation in (a) video surveillance and (d)(e) television or newspaper; (b) image blur in Google Maps street view; (c) image scrambling in video surveillance.	6
2.2	P3 privacy protection algorithm (a) and photo sharing architecture (b).	9
3.1	Workflows of JPEG encoding, decoding and transcoding.	16
4.1	Workflow of JPEG Scrambling protection and recovery in (a) JPEG encoding/decoding or (b) JPEG Transcoding.	24
4.2	Example image scrambled in different strength levels.	25
4.3	The syntax of scrambled JPEG image file.	26
4.4	Storage overhead of JPEG Scrambling on different image datasets.	29
4.5	Face recognition results obtained on the original and different protected images using three different recognition methods: Eigenfaces, Fisherfaces and LBPH.	31
4.6	Results of subjective experiment on (a) face recognition and (b) license plate recognition: proportion of “I don’t know”, incorrect and correct answers for original and protected images.	34
4.7	Example license plate protected by JPEG Scrambling and P3 with different parameters.	35
5.1	Overview of JPEG Transmorphing: the protection and reconstruction procedures.	38
5.2	Illustration of a protection procedure of JPEG Transmorphing.	39
5.3	The syntax of Transmorphed JPEG image file.	42
5.4	Block diagram of DCT re-quantization for overhead control in JPEG Transmorphing.	42
5.5	Illustration of DCT coefficients Cut-off with a CF of 5.	43
5.6	Illustration of an image protected by JPEG Transmorphing without and with overhead controls.	44
5.7	Illustration of a reconstruction procedure of Secure JPEG Transmorphing.	44
5.8	Storage overhead of JPEG Transmorphing with overhead controlled by DCT Re-quantization.	47

## List of Figures

---

5.9	Storage overhead of JPEG Transmorphing with overhead controlled by DCT Cut-off. . . . .	48
5.10	Storage overhead of JPEG Transmorphing for images compressed with different JPEG quality factors. . . . .	49
5.11	Illustration of three different image ROIs considered as protection targets. . . . .	49
5.12	Reference and evaluation image sets of an example identity. . . . .	53
5.13	Screenshot of an HIT presenting an image under subjective privacy evaluation on AMT. . . . .	54
5.14	Proportion of “I don’t know”, incorrect and correct answers across all images, with respect to different protection methods and regions. “H”, “U” and “F” annotated on each bar indicates Head, Upper-body and Full-body respectively. . . . .	55
5.15	Certainty scores of correct and incorrect recognition answers with respect to different scenarios and protection methods. . . . .	56
5.16	Certainty scores of correct recognition answers with respect to different scenarios, protection ROIs and protection methods. . . . .	57
5.17	13 images used in pleasantness evaluation of privacy protection methods. . . . .	58
5.18	10 different visual privacy protection methods. . . . .	59
5.19	Overall perception pleasantness scores of different protection methods. . . . .	60
5.20	Distribution of MOS across all 13 images for each protection method. . . . .	60
5.21	Histograms of “Dislike”, “Neutral” and “Like” for different protection methods. . . . .	61
5.22	Correlation between perception pleasantness (MOS) and proportion of votes for three different preference options. . . . .	62
6.1	Illustration of two schemes of attribute-based encryption. . . . .	69
6.2	Overview of ProShare architecture for privacy-preserving photo sharing. . . . .	72
6.3	Screenshots of ProShare iOS application. . . . .	78
6.4	Performance evaluation of ProShare prototype application. . . . .	80
7.1	Workflow of ProShare S architecture. . . . .	82
7.2	Workflow of user study using ProShare S. . . . .	86
7.3	Distribution of (a) images in each category, (b) subjects sharing decisions and (c) images in each location type. . . . .	87
7.4	Performance of decision making at different sizes of training sets. . . . .	89
7.5	Performance of cost-sensitive decision making with two different values of $c$ . . . . .	90
7.6	Performance of a One-Size-Fits-All classifier on decision making. . . . .	91
7.7	Histogram of photo sharing decisions distinguished by different features. . . . .	92
7.8	Correct decision rates obtained on different combinations of features. . . . .	92
7.9	Correct decision rates obtained on combinations of all Image Semantic Features ( $\mathbf{I}_{All}$ ) and different Requester Contextual Features ( $\mathbf{R}$ ) for five example users. . . . .	93
8.1	GIF vs. JPEG. . . . .	100




---

8.2	Syntax of GIF file format. . . . .	102
8.3	Syntax of aJPEG file format . . . . .	103
8.4	Nine GIF images selected from TGIF dataset. . . . .	106
8.5	Six video sequences from the EPFL-PoliMI dataset. . . . .	107
8.6	Normalized aJPEG file size compared to GIF. . . . .	108
8.7	Comparison of PSNR between aJPEG and GIF. . . . .	109
8.8	Comparison of SSIM between aJPEG and GIF. . . . .	110
9.1	An example image processed by seven different manipulations. . . . .	113
9.2	Boxplot of overall VA scores for each image manipulation method. . . . .	116
9.3	$\Delta$ VA scores for different image content and manipulations. . . . .	117
9.4	Scatter plot of all $\Delta$ VA scores due to “Old paper” manipulation. . . . .	117
9.5	Average emotion distribution of different manipulation methods. . . . .	118
9.6	Difference in emotion distributions between original and manipulated images. . . . .	119
9.7	Number of influential factors for different manipulations. . . . .	120
9.8	Framework of an emotion prediction system. . . . .	120
10.1	Image samples of Food-5K dataset. . . . .	128
10.2	Image samples of Food-11 dataset. . . . .	129
10.3	Results of food/non-food classification obtained on evaluation set of Food-5K. . . . .	131
10.4	Confusion matrix of food/non-food classification results on two different image datasets: Food-5K and IFD. . . . .	132
10.5	Misclassified food and non-food images in Food-5K dataset. . . . .	133
10.6	Performance of food categorization on evaluation set of Food-11 dataset. . . . .	134
10.7	Confusion matrix of food recognition. Values of the matrix are in percentage. . . . .	135
10.8	Workflow of a food classification system for dietary assessment. . . . .	135
10.9	Screenshots of prototype Android app for food image classification. . . . .	136
A.1	Screenshot of an HIT on AMT for subjective experiment on face recognition (Chapter 4). . . . .	142
A.2	Screenshot of an HIT on AMT for subjective experiment on license plate recognition (Chapter 4). . . . .	143
A.3	Evaluation images of the six identities in subjective evaluation of different privacy protection methods (Chapter 5). . . . .	144
A.4	Screenshot of an HIT on AMT for subjective experiment on pleasantness of visual privacy protection methods (Chapter 5). . . . .	145
A.5	Application screenshots. . . . .	146
A.6	Screenshots of ProShare S Android application used for user study in Chapter 7: (a) login page, (b) main page showing all photos, (c)-(f) im- age semantics annotation, (g) contextual sharing decision questionnaire, (h) protect and upload image. . . . .	146
A.7	Screenshots of two prototype applications for aJPEG conversion and play- back in Chapter 8. . . . .	147

## List of Figures

---

- A.8 Screenshot of a questionnaire on Microworkers for emotion evaluation of image manipulation in Chapter 9. . . . . 147
- A.9 Top 10 class pairs that are misclassified in food image categorization experiment in Chapter 10. The percentage refers to the proportion of images in evaluation set for a particular category. The symbol  $\neq$  stands for “incorrectly classified as”. . . . . 148



# List of Tables

3.1	Common JPEG markers. . . . .	20
4.1	Results of objective privacy evaluation of JPEG Scrambling and P3: face detection rate, license plate recognition rate and SSIM (mean). . . . .	31
5.1	Mean PSNR (dB) and SSIM scores of images reconstructed from JPEG Transmorphing and P3, without and with image transformations applied. . . . .	50
5.2	Visual privacy protection methods put in comparison in privacy evaluation. . . . .	52
5.3	Visual privacy protection methods being compared in pleasantness evaluation. . . . .	59
5.4	Qualitative comparison of different reversible visual privacy protection methods. Sc., Cr., Co. and Ro. denote four types of image transformations, namely scaling, cropping, JPEG compression and rotation respectively. . . . .	63
6.1	Notations used in describing ProShare architecture. . . . .	71
7.1	Notation and definition of features in ProShare S. . . . .	84
7.2	The cost matrix applied in cost-sensitive learning. . . . .	90
9.1	Features used for predicting evoked emotions upon image manipulation. . . . .	121
9.2	Results of emotion prediction based on 10-fold cross validation. . . . .	122
10.1	Categories, example items and number of images in each subset of Food-11. . . . .	130
10.2	Accuracy of food/non-food image classification on evaluation set of Food-5K for the two fine-tuning configurations. . . . .	131



# 1 Introduction

Thanks to advancements of smart mobile phones and social media platforms, sharing photos and experiences has significantly bridged our lives, allowing us to stay connected despite distance and other barriers. The number of images shared from mobile devices has reached scales which were unimaginable only a decade ago: Every day over two billion images are posted to online social networks (OSNs) or exchanged through instant messaging and cloud-based sharing services. This fact has transitioned the challenges we faced before like improving image quality into challenges such as how to store them, how to preserve them in long term, how to annotate or tag them properly, how to access them efficiently (search and retrieval), how to utilize them to create added value, and of course how to share this huge amount of content in the educational, consumer and professional sectors, not only efficiently but also in trustworthy manners, taking into account security issues, including privacy matters which are involved. Many photo and video sharing social networks or services have developed various features and advantages to help users share their photos or videos more easily and conveniently. This has positioned photo or video sharing in social networks among the most popular and fastest growing applications in the World Wide Web.

Despite an unquestionable value in terms of new experience such applications offer to their users, sharing these media in social networks has created a number of new problems. Most critical issues concern lack of trust and problems regarding privacy of the shared content. In addition, with the latest progress in image analytics, pattern recognition, deep learning, in combination with multi-modal data mining of personal information from mobile and social networks, the world of Orwell's *1984* [1] seems to become a sober reality in the near future, if not already now. Such disconcerting situation with privacy protection does not only result in a growing number of media scandals (leaking of celebrity and politicians private photos), but also hurts normal people, when their exposed personal photos or video affect their work, life, and even health. The number of sharing services has rapidly grown over the past few years making users more concerned about these issues. Although sharing services have assisted resolving these challenges by allowing users to

restrict content access and making them available to specific list of users, it is not in their best interest if users restrict access to their shared pictures and these challenges are still prevailing. The stellar success of Snapchat<sup>1</sup>, an instant photo messaging service, demonstrates the pent-up demand for privacy. Here a sense of privacy is created through an ephemeral service model [2] where shared pictures remain visible for a short period of time after which they “disappear”. Irrespective of a service provider’s sincerity in matters of privacy, all image sharing platforms exhibit the same basic flaw: Once an image has left the device it was created on, its owner loses control over who will have access to his image, when and where.

Researchers have proposed and developed various approaches to enhance privacy in photo sharing. A substantial efforts have been devoted to design of access control protocols in more intelligent or adaptive manner. Essentially, most approaches have no difference with most conditional access mechanisms applied by popular social networking services. Another branch of studies have been focused on methods for secure protection of image content itself, by means of encryption or distortion-based image processing or encoding. An advantage of secure protection of image content over access control lies in the fact that the original content is not even available to internal service providers such that users can leverage less trust on them. However, such approaches usually raise higher requirements for file management system and generate distorted visual effect in protected content, which may compromise the utility, pleasantness and user experience of photo sharing. Such impact has not yet been well understood.

Given today’s social media challenges, a desired method for protecting photo privacy needs to provide the following characteristics: (i) security (powered by state-of-the-art cryptographic tools), (ii) low complexity (fast, easy and intuitive to use), (iii) reversibility (possibility to undo protection), (iv) compatibility (compatible with standard image compression and file format), (v) robustness (possibility to be recovered even after being manipulated), (vi) variable granularity (flexibility to protect data in different regions, portions or with different degrees of strength) and (vii) pleasantness (giving a sense of friendliness or enjoyment or at least not annoying). However, most existing solutions cannot fulfill all the mentioned properties.

In this thesis, we investigate novel solutions to protect image privacy with a special emphasis on the scenario of online photo sharing. Our solutions include not only the algorithms to protect visual information in image but also designs of architectures for privacy-preserving photo sharing. Beyond privacy, we also explore the potentials and additional impacts of using daily images in other three relevant applications. Therefore, the thesis is structured in three parts: the first part on image privacy protection algorithms, the second on photo sharing architectures, and the third on relevant applications beyond. The rest of the thesis is structured in detail as follows.

---

<sup>1</sup><https://www.snapchat.com/>

## 1.1 Thesis Outline

First in Chapter 2, we review related studies on image privacy protection, privacy-preserving photo sharing, and contextual information sharing in general. Then, the main body of the thesis is constructed by the following three major parts.

In Part I, we propose and study two image visual privacy protection algorithms based on JPEG compression. We first provide a brief overview of JPEG compression and a conceptual framework named Secure JPEG in Chapter 3. Then in Chapter 4, we elaborate a scrambling-based algorithm for protecting image visual content, by randomly changing the signs of quantized DCT coefficients of a JPEG image. In Chapter 5, we propose the second method named JPEG Transmorphing. Designed in a different philosophy from any existing privacy protection algorithm, JPEG Transmorphing allows one to obfuscate arbitrary image region(s) while secretly preserving the original information corresponding to that region(s) in application segments of the obfuscated JPEG image. Therefore, JPEG Transmorphing is not restricted to any type of visual obfuscation. Instead, most regional image manipulations can be applied to protect visual privacy within the framework of JPEG Transmorphing. We conducted objective and subjective experiments to evaluate the performance of both methods in regard to different aspects, including the storage overhead created, reconstruction quality, privacy protection capability and pleasantness.

In Part II, we investigate two different photo sharing architectures with privacy protection in mind. Chapter 7 presents the first architecture named ProShare, designed based on a public key infrastructure (PKI) with a ciphertext-policy attribute-based encryption (CP-ABE) integrated. We implemented and demonstrate the ProShare architecture in both the iOS and Android mobile platforms. Chapter 7 presents the second architecture named ProShare S, a photo sharing decision making system by analyzing image semantics and context information based on machine learning. We conducted a user study along with extensive performance evaluations to validate the ProShare S architecture.

In Part III, we research into three relevant topics in regard to daily images captured or shared by people, but beyond their privacy implications. Chapter 8 adopts the idea from JPEG Transmorphing and presents an animated JPEG file format, called aJPEG, which could serve as a better alternative to conventional GIF. Chapter 9 attempts to understand the influence of popular image manipulations applied in online photo sharing on evoked emotions of photo observers. By learning from image features such as color and texture, we build and evaluate a classifier that can accurately predict the emotions of a manipulated image given as input only the original image and the desired manipulation. In the last study (Chapter 10), we target on dietary assessment using daily images captured by people. To this end, we employ a deep convolutional neural network (CNN) to perform automatic food image detection and categorization, as the initial but key step to automatic dietary assessment based on multimedia techniques, e.g. image analysis. Finally, Chapter 11 summarizes the thesis and discusses future work.

## 1.2 Summary of Contributions

The main contributions of this thesis are summarized as follows:

- We propose a novel encoding/transcoding algorithm for protecting regional visual privacy in JPEG image, named JPEG Transmorphing. We show the significant advantages of using the proposed algorithm for privacy protection, in terms of storage overhead, reversibility, privacy protection capability and particularly pleasantness in both perception and usage perspectives.
- We propose and demonstrate an architecture for privacy-preserving photo sharing, ProShare, based on the proposed Secure JPEG privacy protection algorithms. The architecture employs a combination of traditional public key infrastructure (PKI) and cipher-policy attribute based encryption (CP-ABE) to enable conditional access to original images in a secure and efficient way. Particularly, using the ProShare architecture, protected images can be safely stored on any “untrusted” server.
- We propose and study another conceptual architecture (ProShare S) for privacy-preserving photo sharing based on analyzing image semantics and access context using machine learning. This architecture (semi-)automatically makes photo sharing decisions based on not only the content of image but also the contextual information about the requester and image capture. This is the first attempt to understand photo sharing privacy preferences in context-dependent way. We conducted a user study and contribute a personalized dataset with user-annotated image semantic features and contextual sharing decisions.
- Inspired by our JPEG Transmorphing algorithm, we propose aJPEG, a new animation image format within the framework of JPEG compression. aJPEG encodes a default frame of an animation sequence, while compressing and preserving the other frames in application segments of the default frame JPEG image. This format proved to offer smaller file size and higher image quality compared to conventional GIF, and therefore could serve as a better alternative to GIF.
- We investigate the influence of popular image manipulations applied in photo sharing on evoked emotions as perceived by people. This study reveals that image content and applied manipulation both affect the evoked emotion. In addition, we create and study a regressor that can accurately predict the expected emotions of a prospective manipulated image given as input only the original image and the desired manipulation to apply.



## 2 State of the Art

Early approaches to image or video privacy protection mainly aim to protect personal privacy in video surveillance or television. Typically, the goal is to distort video or image, remove or hide visual information, which can be used for people and location identification or any sensitive information disclosure (see Figure 2.1 for examples). Recent popularization of online social media (e.g. photo sharing) has raised different requirements for privacy protection due to its different nature compared to video surveillance. Researchers in fields of image processing, multimedia, security and cryptography have proposed different solutions to enable privacy for different use cases, including video surveillance and social media. These solutions are different in terms of working principle, complexity, effectiveness of the privacy protection, reversibility, usage flexibility and pleasantness. We review some of them in this chapter.

### 2.1 Privacy Protection in Image Visual Content

Since visible identifiable face is a major threat to privacy, many researchers have focused on face de-identification techniques. For instance, in [3] people's identities are protected by obscuring their face with a colored ellipse. The authors argue that such protection allows observation of the people actions in full details while hiding their identity. Other naïve approaches also include blurring and face masking for hiding the faces of the people in the video. Arguing that de-identification of faces is not enough for an adequate privacy protection, the technique for obscuring of the whole body silhouette is proposed in [4], which is based on the edge and motion model. Going further, in [5] and [6] it is proposed to completely remove the silhouette of the moving person from the scene to hide its identity. Both approaches rely on RFID tags for pinpointing of the people locations, with [6] focusing on an efficient inpainting algorithm and encrypting the removed silhouette inside the original video bitstream. However, all these filters irreversibly distort video data at the pixel level, making it impossible to use video in situations, when, for example, due to a court order, an identity of the hidden person needs to be revealed.

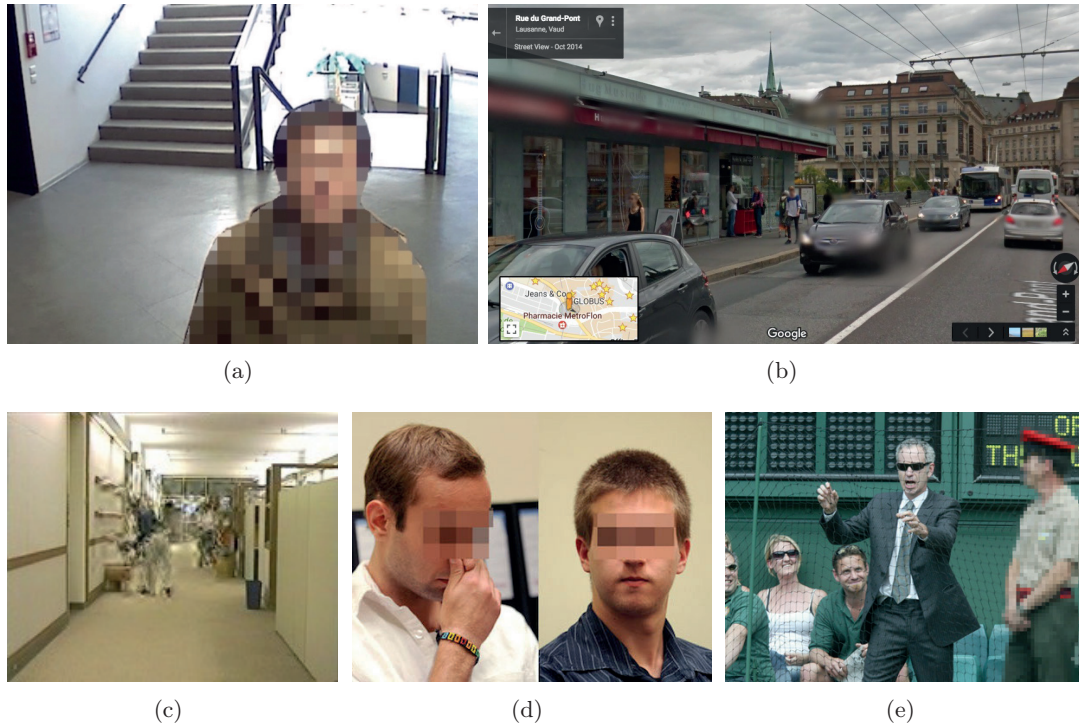


Figure 2.1 – Examples of different visual privacy protection approaches: image pixelation in (a) video surveillance and (d)(e) television or newspaper; (b) image blur in Google Maps street view; (c) image scrambling in video surveillance.

Aiming to avoid constraints of the distortion-based methods, more advanced scrambling-based privacy filters are proposed in [7, 8] and [8]. These techniques are based on randomized (seeded with a secret key) modifications of the compressed video stream encoded as a series of JPEG and JPEG 2000 images. The main advantage of scrambling-based privacy protection techniques is that they are reversible. By knowing a secret key, which could be stored and transmitted securely, one can decode the video back to undistorted state. Another advantage is that the appearance of the scrambled regions are not completely distorted, making the viewing experience less distracting (compared to a black box covering an area of the picture for instance).

Another way to protect a sensitive region securely is to encrypt it. With a system named PICO proposed in [9], data corresponding to face regions is encrypted in order to conceal identity. The process is reversible for authorized users in possession of a secret encryption key. Similarly, a permutation-based encryption technique in the pixel domain is introduced in [10]. TrustCam is presented in [11], which is a video camera with onboard hardware security solution Trusted Platform Module (TMP), which implements trusted computing. This built-in chip allows establishing secure connection between cameras and observing stations, as well as applying SHA-1 based encryption to the sensitive regions,

---

## 2.2. Privacy Protection in Online Photo Sharing

such as faces and license plates. The idea of encrypting or scrambling face regions was developed further by [12], where the authors argue that the conventional encryption methods are not suitable due to the real-time constraints, limited computational and network resources. Instead, they suggest adding a special parameter inside an encoder compression block, which would enable encryption and secret key generation.

A substantial number of studies have focused on encryption-based protection of JPEG images, as JPEG is the most widely used image format. As the core of JPEG compression is Discrete Cosine Transform (DCT), most methods encrypt the transformed image data, namely the DCT coefficients of a JPEG image, in different ways. Niu et al. [13] propose a JPEG Encryption scheme without significantly increasing the image file size. In the proposed method, the DC differential residues are encrypted through XOR with a secure key of the same length as the data stream. DCT blocks are scrambled using a key-controlled chaotic map and the information of pre-steps is encrypted by cipher and embedded in the second category of AC coefficients. Unterweger [14] proposes a method to encrypt baseline JPEG bit streams by swapping selective Huffman code words and scrambling DCT coefficient values based on AES encryption. Wright et al. [15] propose a special image encryption scheme that permutes pixels in each image block individually. The proposed technique allows efficient reconstruction of an accurate low-resolution thumbnail from the ciphertext image, but aims to prevent the extraction of any more detailed information. This will allow efficient storage and retrieval of image data in the cloud but protect its contents from outside hackers or snooping cloud administrators. Recently, [16, 17, 18, 19] propose different encryption protocols to encrypt the quantized DCT coefficients of JPEG image and all these methods proved to well support common image manipulations performed by social networking sites such that decrypted images are still highly similar to the original image.

## 2.2 Privacy Protection in Online Photo Sharing

Most social networks provide users with some privacy access control mechanisms, which are essentially a set of ad hoc rules that restrict the access to users content, thus creating an illusion of privacy protection. The stellar success of Snapchat<sup>1</sup>, an instant photo messaging service, demonstrates the pent-up demand for privacy. Here a sense of privacy is created through an ephemeral service model where shared pictures or video remain visible for a short period of time after which they “disappear”. The research studies on solutions to privacy protection in social media, especially for online photo sharing, are mainly focused on two directions: (i) design of access control protocols such that the shared photos can only be accessed by a selected group of users; (ii) algorithms for secure protection of image content (encryption, scrambling, permutation, etc.) while photo sharing. We outline selected studies on the two directions respectively in the following.

---

<sup>1</sup><https://www.snapchat.com/>

### 2.2.1 Access Control

Squicciarini et al. [20, 21] propose an Adaptive Privacy Policy Prediction (A3P) system to help users compose privacy settings for their images. In their method, image content and metadata are examined by indicators of users privacy preferences. A two-level image classification framework is created to obtain image categories which can be associated with similar policies. Then a policy prediction algorithm that can automatically generate a policy for each newly uploaded image is designed. Most importantly, the generated policy can follow the trend of the user's privacy concerns evolved with time. Cutillo et al. [22] propose a preliminary usage control mechanism targeting decentralized peer-to-peer online social networks where control is enforced thanks to the collaboration of a sufficient number of legitimate peers. In the proposed solution, all faces in image are automatically obfuscated when being upload to the system and the enforcement of the obfuscation operation is guaranteed by the underlying privacy-preserving multi-hop routing protocol. The disclosure of each face depends on the rules set by the face owner when she is informed and malicious users can never publish this content in clear even if they have access to it. In [23], the feasibility of using image tags to create effective access-control rules is studied. The study conducted a subjective user study and results reveal that organizational tags can be repurposed to create reasonable access-control policies, and that policies based on these tags are yet more accurate when subjects actively create tags for access control. The paper suggests that it would be possible to create a usable access-control system with tag-based rules and minimal tagging overhead. It may be possible to additionally aid users with appropriate support for automated rule generation, exception handling, intuitive policy management, and automated tag generation and correction. Recently, Lee et al. [24] propose a fine-grained multiparty access control model, which aims to change the granularity of access control from photo level to face level. The proposed model evaluates the policy of each user recognized in given photo, based on relationship intimacy, photo context about spatial-temporal information and co-occurrence users, and finally generates policy for each face. Similarly, Ilia et al. [25] propose a system that allows users to effectively prevent unwanted individuals from recognizing faces in a photo. The core concept behind the proposed approach is to change the granularity of access control from the level of entire image to that of a user's personally identifiable information (PII), i.e. the face in the case of the study. When another user attempts to access a photo, the system determines which faces in image the user does not have the permission to view, and then presents the photo with the restricted faces blurred out. In addition to the above, a great number of studies have been carried out for enhancing privacy in social networks, e.g. Persona [26], EASiER [27], NOYB [28], FlyByNight [29] and Lockr [30], most of which rely on encryption protocols to enable access control in social networks or secure sharing of general data (status, message, profile, image, etc). Though not particularly focused on privacy of photo content sharing, these studies also closely relate to our design of photo sharing architecture.

## 2.2. Privacy Protection in Online Photo Sharing

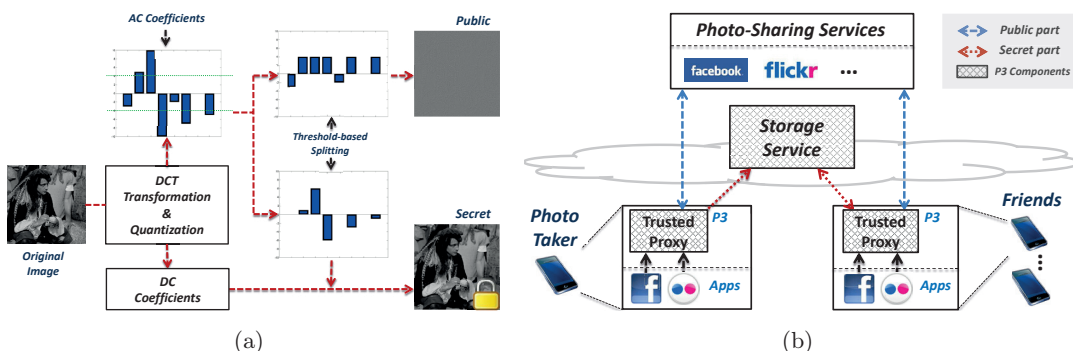


Figure 2.2 – P3 privacy protection algorithm (a) and photo sharing architecture (b).

### 2.2.2 Secure Protection of Image Content

Another groups of methods focus on secure protection of image content itself to able privacy from a content level in photo sharing. Poller et al. [31] present two approaches to robust image obfuscation based on permutation of image regions and channel intensity modulation both in image spatial domain. The proposed approaches take into account the fact that images uploaded to Web 2.0 applications pass several transformations, such as scaling and JPEG compression, and therefore enable the reconstruction of unprotected images in high quality. Instead of providing the maximum of security, the proposed methods focus more on usability and try to obtain an acceptable trade-off between security and resulting image quality. Ra et al. [32] propose a privacy-preserving photo encoding algorithm named P3 that extracts and encrypts a small, but significant, component of the photo, while preserving the remainder in a public and standards-compatible part. These two components can be separately stored. The P3 technology significantly reduces the accuracy of automated detection and recognition on the public part, while preserving the ability of the provider to perform server-side transformations to conserve download bandwidth usage. The prototype privacy-preserving photo sharing system powered by P3 works with Facebook, and can be extended to other services as well. However, the major drawback of P3 is that the protected public portion can still reveal certain degree of visual information, which might compromise privacy. The encoding protection process of P3 and a secure photo sharing architecture based on P3 is illustrated in Figure 2.2. Tierney et al. [33] propose a system named Cryptagram, which enables users to encrypt photos with traditional block ciphers and embed the encrypted bitstream into a JPEG “cover image”. However, the proposed approach creates a significant expansion to image file size due to the use of a cover image. This will impact the system usability. Zhang et al. [34] introduce a framework called POP, which enables privacy-seeking mobile device users to outsource burdensome photo sharing and searching safely to untrusted servers. With a carefully designed architecture and novel non-interactive privacy-preserving protocols for image similarity computation, unauthorized parties, including the server, learn nothing about photos or search queries. For efficiency and good user experience, the proposed

framework allows users to define personalized private content by a simple check-box configuration and then enjoy the sharing and searching services as usual. All privacy protection modules are transparent to users. However, this study focuses more on the problem of image searching or retrieval instead of photo sharing. In addition, [18, 19] and [35] all demonstrate the feasibility of using their proposed encryption methods to protect privacy in JPEG image on popular social networks, e.g. Facebook. The results show that the encrypted image can still be decrypted even if it has been manipulated on social network using common image processing like image scaling and JPEG compression.

### 2.3 Privacy Analysis in Image

Research efforts have also been devoted in identifying the potential privacy thread of sharing image content, understanding users privacy concern, automatic recognition of privacy-sensitive image regions, or even trying to break existing privacy protection solutions. All these studies mainly aim at providing insights, implications or suggestions to build stronger, more secure and usable privacy protection solutions. We call all these, collectively, *privacy analysis*. Early in 2007, Ahern et al. [36] conducted a qualitative and quantitative analysis of privacy in a real-world photo-sharing mobile and online application, to understand users privacy patterns and considerations in online and mobile photo sharing. In their study, context-aware cameraphones were used as capture devices which allow us to conduct a subjective user study. This is one of the first study that tries to understand people privacy concern and behavior in online photo sharing. Besmer et al. [37] examine privacy concerns and mechanisms surrounding tagged images in social networking environment that provides photo tagging feature. The authors explore the needs and concerns of users and propose a set of design considerations for tagged photo privacy. Friedland et al. [38] make a case for the emerging privacy issue caused by wide-spread adaptation of location-enabled photo and video capturing devices, allowing potential attackers to easily “case out joints” in cyberspace. The aim of this study is to raise awareness of a rapidly emerging privacy threat cause by geo-tagging in images and calls for research effort on designing systems to be location-aware while at the same time offering maximum protection against privacy infringement. Pesce et al. [39] expose some of the privacy issues with photo albums, especially the use of photo tags to predict information about Facebook users. The aim of the study is to show that the use of photo tagging can enhance accuracy of attackers aiming to predict personal user attributes and to raise awareness of the kinds of information transmitted by photo tags in social networks, thus avoiding collateral damages. Zerr et al. [40] propose techniques to automatically detect private images using machine learning, and to enable privacy-oriented image search. In this study, privacy classifiers are trained on a large set of manually annotated Flickr photos, combining textual metadata of images with a variety of visual features. The classification models can be used for searching for private photos, and for diversifying query results to provide users with a better coverage of private and public content.

Recently years, researchers started to use the latest deep convolutional neural networks (CNNs) to identify privacy sensitive objects or regions in images. This types of studies include [41, 42, 43], most of which utilize different CNN models for privacy-sensitive region detection and recognition. In addition, both [44, 45] employ deep learning approaches to defeat common image obfuscations such as image blurring, masking and P3 [32]. The two studies reveal that deep learning can be used to accurately recognize some faces, objects and handwritten digits even in visually obfuscated images. The results reveal privacy implications of photo sharing even with certain visual protection applied in image, and provide significant insights in designing stronger privacy protection methodologies that can both enable privacy and preserve the maximum usability.

## 2.4 Context-Dependent Information Sharing

When sharing photo information, users may take into account several factors to balance privacy, utility and convenience. The factors include the information about the image content and different contextual information such as location, time, activities and presence of other people. In this section, we review the studies on contextual and context-dependent information sharing.

A substantial research efforts have been made to understand users behavior and privacy attitudes towards online information sharing, and the factors that influence their decisions. Smith et al. [46] provide an early investigation on solutions to enable people to share contextual information (e.g. location) in mobile social networks. The authors developed a system called Reno that can automate the decision making process of location sharing, based on a set of pre-defined regions. However, they show that static rules for location sharing in pre-defined regions are not accurate enough in expressing the users actual behavior when other contextual information changes, such as the time. Toch et al. [47] study the influence of the type of locations visited by users on their willingness to share the locations with others. Simple statistical models reveal that the semantic category of the location being shared (such as a shopping center or a hospital) and the social group of the person asking for the location are significant factors in location information decision making. These results also agree with early studies in [48, 49, 50] in providing the most influential contextual features for location-sharing. We use these results in our system design for context-dependent photo sharing (Chapter 7). Benisch et al. [51] compare simple access control policies with more sophisticated ones (based on time, day and location) and find out that the accuracy of the sharing policies increases as their complexity (or flexibility) increases. Besides, they also observe that the accuracy benefits are the most notable for the information that is highly sensitive. This suggests that the cost of incorrect information sharing (to unauthorized parties) is an important factor in designing and optimizing automated information-sharing mechanisms. This concept is also used in our study in a way of designing a cost-sensitive decision making mechanism in online photo sharing (Chapter 7). Wiese et al. [52] investigate the impact of various

factors on users willingness to share information. The results of the study reveal that social closeness and the frequency of communication perform as better predictors of sharing than physical proximity and social groups of the people asking for the information. The authors suggest that automatic methods for inferring social closeness could be suited for automatic information-sharing decisions more than physical co-location.

In addition to the studies on photo sharing policy inference [20, 21, 23, 25, 24] introduced in Section 2.2.1, decision making for general information sharing has also been widely studied and most of them are based machine learning approaches. Sadeh et al. [50] compare the accuracy of user-defined sharing policies with an automated mechanism and a machine learning based approach (random forest). Results reveal that automated approaches have a better accuracy than those user-defined policies, and provide insights of applying machine learning in information sharing decision making systems. Fang et al. [53] propose an approach to infer access control policies for personal information on online social networks, based on supervised-learning. The learning procedure is done by asking each user a limited number of questions about her/his sharing behavior with specific friends. Bigwood et al. [54] evaluate the performances of different machine-learning algorithms for predicting information sharing decisions in terms of correct decisions and information over-sharing. In their study, cost-sensitive classifiers are also used to reduce over-sharing cases. However, they only focuses on a binary (yes/no) location-sharing problem. Xie et al. [55] study the influence of different contextual (e.g. semantics of the location) and personal features on users location-sharing behaviors. A recommendation system for privacy preferences is proposed in this study and the system determines the recipients to whom a given piece of information can be shared. Recently, Harkous et al. [56] present a conceptual framework named C3P for automatic estimation of privacy risk of data based on the sharing context. The framework lets users crowdsource their sharing contexts with the server and determine the risk of sharing particular data item(s) privately, thus helping users make decisions in information disclosure. As a study most related to ours, Bilogrevic et al. [57] present SPISM, an information-sharing system that predicts (semi-)automatically sharing decision, based on personal and contextual features. However, they focus on only general information sharing such as location, nearby people and availability. Despite the substantial works on contextual information sharing, very few have considered contextual information for privacy protection in online photo sharing.



# The Algorithms **Part I**



# 3 Overview of JPEG and Secure JPEG

The success of digital imaging applications is in part due to the development of effective image compression standards such as JPEG. JPEG is one of the early standards and is de facto the most popular compression standard to store or transmit images thanks to its efficiency and low complexity. It has remained the most popular image format used in a large variety of consumer imaging applications, e.g. digital camera, smartphone and social media. In this thesis, the proposed methods for protecting image privacy are designed on the basis of JPEG compression and protected images are required to be backward compatible with JPEG. Such an idea was initially proposed in the form of Secure JPEG [58], an open and flexible framework to secure JPEG images. This chapter provides an overview of JPEG compression, its file format and the Secure JPEG framework, which are closely related to the design of our algorithms and architectures in the thesis.

## 3.1 Overview of JPEG

JPEG is the most commonly used method of lossy compression for digital images, especially for those images produced by digital photography. JPEG compression is used in a number of image file formats, e.g. JPEG/Exif (the most popular image format used by digital cameras or other photographic capture devices) and JPEG/JFIF (the most common format for storing and transmitting photographic images on the World Wide Web<sup>1</sup>). We usually do not distinguish these format variations and collectively call them JPEG. The term “JPEG” is originally an acronym for the Joint Photographic Experts Group, a joint committee between ISO/IEC JTC1<sup>2</sup> and ITU-T<sup>3</sup>, which created the JPEG compression standard. This section provides a brief overview of JPEG compression, including its encoding, decoding, transcoding processes and the file formats and syntax.

---

<sup>1</sup><http://httparchive.org/interesting.php#imageformats>

<sup>2</sup><https://www.iso.org/isoiec-jtc-1.html>

<sup>3</sup><http://www.itu.int/>

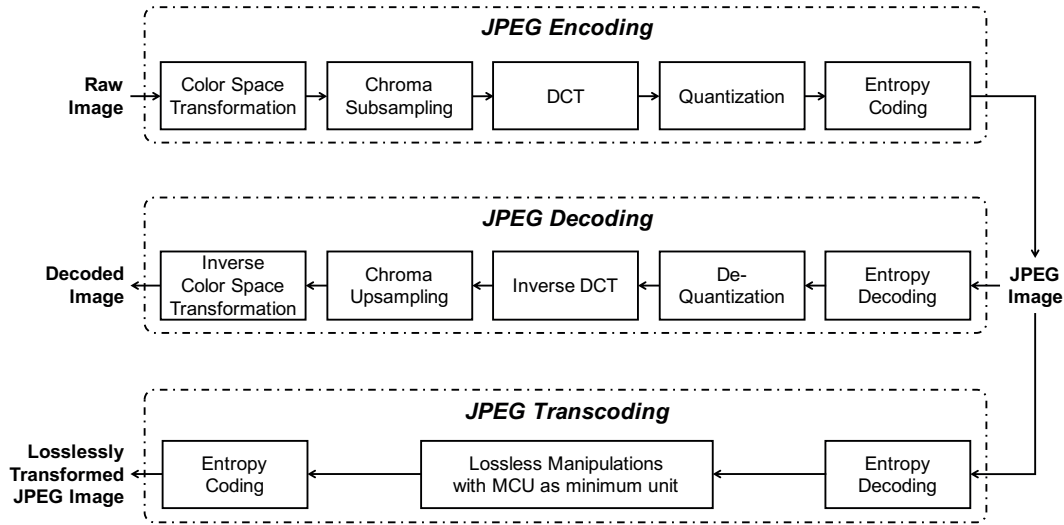


Figure 3.1 – Workflows of JPEG encoding, decoding and transcoding.

### 3.1.1 JPEG Compression

First of all, Figure 3.1 illustrates the workflows of JPEG encoding, decoding and transcoding respectively. Then the three processes are described in detail as follows.

#### JPEG Encoding

JPEG encoding, or compression process, aims at compressing the raw image data into a compressed form to reduce irrelevance and redundancy of the image data. An entire JPEG compression procedure usually consists of the following steps:

**Color Space Transformation** The first step is to convert the raw image data from the RGB color space to the YCbCr color space, where Y channel represents the luminance (brightness) component and Cb/Cr the chroma components representing color. YCbCr can be computed directly from 8-bit RGB according to the following formula:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.0813 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}. \quad (3.1)$$

**Chroma Subsampling** Since human can see more fine details in the brightness of an image (the Y channel), than in the hue and color saturation (the Cb and Cr components), the two color channels are usually subsampled at a lower resolution than the Y channel to reduce the amount of information to be encoded without significantly affecting the perceptual quality. Typical subsampling options include 4:4:4 (no subsampling), 4:2:2 (downsampling by a factor of 2 in the horizontal direction), and the most commonly used

4:2:0 (downsampling by a factor of 2 in both horizontal and vertical directions).

**Discrete Cosine Transform (DCT)** After subsampling, each channel of the image is divided into non-overlapped blocks, each of  $8 \times 8$  pixels. For an 8-bit image, every pixel value in each channel falls in the range of  $[0, 255]$ . The midpoint of the range (i.e. 128 in this case) is subtracted from each pixel to produce a data range of  $[-128, 127]$  centered on zero. Then the 2D Discrete Cosine Transform (DCT) is applied to each block, resulting in  $8 \times 8$  DCT coefficients. The first value of the 64 DCT coefficients is called the DC coefficient, which represents the mean of all the pixels within a block. The remaining 63 coefficients are called AC coefficients, representing intensity changes across the block. The DCT coefficients represent the information about the image block in the frequency-domain. Given an image  $x$  in the spatial domain and its pixel at coordinates  $(n_1, n_2)$  denoted as  $x_{n_1, n_2}$ , a 2D DCT is given by the formula:

$$X_{k_1, k_2} = \frac{1}{4} C_{k_1} C_{k_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_1, n_2} \cos \left[ \frac{\pi}{N_1} \left( n_1 + \frac{1}{2} \right) k_1 \right] \cos \left[ \frac{\pi}{N_2} \left( n_2 + \frac{1}{2} \right) k_2 \right], \quad (3.2)$$

where  $k_1 = 0, 1, 2, \dots, N_1 - 1$  and  $k_2 = 0, 1, 2, \dots, N_2 - 1$  and  $N_1$  and  $N_2$  are the dimensions of a DCT block, i.e.  $N_1 = N_2 = 8$ . In Equation 3.2,  $C_{k_1}$  and  $C_{k_2}$  are two normalization constants defined as

$$C_{k_i} = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } k_i = 0 \\ 1, & \text{otherwise} \end{cases}, \quad i = 1 \text{ or } 2. \quad (3.3)$$

**Quantization** Then, all the float-valued DCT coefficients are quantized to integers. This is the only step that causes image information losses in the entire compression procedure. Due to the fact that human eyes are more sensitive to small variances in the low-frequency than high-frequency image regions and that image information is usually concentrated in the low- to medium-frequency components, larger quantization steps are applied in the high-frequency DCT coefficients while smaller quantization steps in lower-frequency coefficients. To this end, a quantization table is employed to represent different quantization steps for different coefficients. The standard quantization tables defined by Independent JPEG Group (IJG)<sup>4</sup> for luminance (Y) and chrominance channels

---

<sup>4</sup><http://www.ijg.org/>

(Cb and Cr) are given in the following respectively:

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 33 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \& \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}. \quad (3.4)$$

A quality factor (Q) in range of 1 to 100 is used to scale the values in the above standard quantization tables to generate scaled quantization tables, achieving different compression ratios. Q = 100 corresponds to the highest quality while 1 to the lowest. Note the standard quantization table(s) as  $\mathbf{T}_b$ , then the scaled quantization table(s)  $\mathbf{T}_s$  are given by the following formula:

$$\mathbf{T}_s = \frac{s \times \mathbf{T}_b + 50}{100}, \quad (3.5)$$

where  $s$  is a scale factor define as

$$s = \begin{cases} 5000/Q, & \text{if } Q < 50 \\ 200 - 2 \times Q, & \text{otherwise} \end{cases}. \quad (3.6)$$

**Entropy Coding** As the last step, quantized DCT coefficients are encoded with a run-length encoding (RLE) algorithm, which groups similar frequencies together, inserting length coding zeros, and then using Huffman coding on what is left. The 64 DCT coefficients are scanned in a zigzag order while coding, to group lower-frequency coefficients in top of data vector with higher-frequency coefficients at the bottom. The difference between two consecutive DC coefficients is encoded rather than the actual values. While the 63 quantized AC coefficients do not apply such a prediction-based scheme. This is called baseline sequential encoding, which encodes coefficients of a single block at a time in a zigzag manner. JPEG also supports progressive encoding, which encodes similar-positioned batch of coefficients of all blocks in one go (called a scan), followed by the next batch of coefficients of all blocks, and so on. The advantage of progressive JPEG is that one can see an approximation to the whole image very quickly while an image is being viewed on-the-fly as it is transmitted. The quality is gradually improved as one waits longer. However, each scan of a progressive decoding takes about the same amount of computation to display as a whole baseline JPEG file would. So the progressive encoding is not often used. In addition to Huffman coding, the JPEG standard also supports arithmetic coding, which typically makes files about 5 ~ 7% smaller but has rarely been used due to its lower efficiency and royalty issues.

### JPEG Decoding

The decoding process of a JPEG image follows the inverse of the above encoding operations. The quantized DCT coefficients are first decoded with the entropy decoder. The DCT coefficients are then dequantized by multiplying the original values with the entries in quantization tables extracted from the JPEG file header. The decoder then applies on each DCT block the Inverse Discrete Cosine Transform (IDCT) to produce the subsampled YCbCr image channels. The IDCT is defined as

$$x_{n_1, n_2} = \frac{1}{4} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} C_{k_1} C_{k_2} X_{k_1, k_2} \cos \left[ \frac{\pi}{N_1} \left( n_1 + \frac{1}{2} \right) k_1 \right] \cos \left[ \frac{\pi}{N_2} \left( n_2 + \frac{1}{2} \right) k_2 \right]. \quad (3.7)$$

Then the channels are upsampled to original image size, shifted by 128 and converted back to RGB color space according to the following formula:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.34414 & -0.71414 \\ 1 & 1.772 & 0 \end{bmatrix} \begin{bmatrix} Y \\ Cb - 128 \\ Cr - 128 \end{bmatrix}. \quad (3.8)$$

### JPEG Transcoding

Certain image manipulations can be applied on a JPEG image without information loss, by directly operating the JPEG Minimum Coded Unit (MCU) blocks while keeping the information in each MCU intact. An MCU block contains several DCT blocks of the YCbCr channels, usually with 16 pixels in both directions, for a 4:2:0 chroma subsampling. These operations include image rotation (in 90-degree increments), flipping (in horizontal, vertical or diagonal axes) and cropping (with MCU block as the minimal unit). The concept of JPEG lossless transcoding is an important basis for the design of our secure protection algorithms, which also directly manipulate as unit of MCU or DCT blocks of certain regions in an existing JPEG image, without altering other DCT coefficients outside those regions.

#### 3.1.2 JPEG File Format and Syntax

Image files that employ JPEG compression are collectively called JPEG files, and are stored in variants of the JPEG interchange format defined in the JPEG standard (ITU-T Recommendation T.81 | ISO/IEC10918-1 [59]). A JPEG image file consists of a sequence of segments, each beginning with a marker. Each marker starts with a 0xFF byte followed by a byte indicating the type of the marker. Some markers consist of just those two bytes; others are followed by two bytes indicating the length of marker-specific payload data that follows. Consecutive 0xFF bytes can be also used as fill bytes for padding purposes. Several common JPEG markers are listed in Table 3.1.

Table 3.1 – Common JPEG markers.

Marker	Bytes	Description
SOI	0xFF 0xD8	Start Of Image
DHT	0xFF 0xC4	Defines one or more Huffman Table(s)
DQT	0xFF 0xDB	Defines one or more Quantization Table(s)
APPn	0xFF 0xEn	Application-specific markers, e.g. APP0 for JFIF, APP1 for Exif
SOF0	0xFF 0xC0	Start Of Frame (baseline DCT): Indicates that this is a baseline DCT-based JPEG, and specifies the width, height, number of components, and component subsampling.
SOF2	0xFF 0xC2	Start Of Frame (progressive DCT): Indicates that this is a progressive DCT-based JPEG, and specifies the width, height, number of components, and component subsampling
SOS	0xFF 0xDA	Start Of Scan: Begins a top-to-bottom scan of the image. In baseline DCT JPEG images, there is generally a single scan. Progressive DCT JPEG images usually contain multiple scans. This marker specifies which slice of data it will contain, and is immediately followed by entropy-coded data.
EOI	0xFF 0xD9	End Of Image

**JPEG Application Segments** JPEG provides a set of application-specific markers, i.e. Application Segments or APPn markers, to specify different variants or extensions of the standard JPEG interchange format. Two typical examples are the JPEG File Interchange Format (JFIF, specified by APP0) [60] and Exchangeable image file format (Exif, specified by APP1) [61]. JFIF is a minimal file format which enables JPEG bitstreams [59] to be exchanged between a wide variety of platforms and applications. It solves some limitations of basic interchange format in regard to simple JPEG encoded file interchange. Most image capturing devices (e.g. digital camera, mobile phone) actually create image files in the Exif format, which has been standardized for interchanging metadata of image and audio files recorded by digital cameras. Those metadata includes capture time, camera model, aperture and shutter setting, number of pixels and even a thumbnail image. Both formats use the actual standard JPEG syntax and are fully compatible with JPEG. They employ different APPn markers of JPEG: JFIF uses APP0 and Exif uses APP1. Strictly speaking, the JFIF and Exif standards are not compatible between each other because each specifies that its marker segment appears first. However, in practice, most JPEG files contain a JFIF marker (APP0) followed by an Exif header (APP1). This allows legacy JPEG decoders to correctly handle the older JFIF format, while newer decoders can also read the following Exif segment, being less strict about requiring it to appear first. Our design of privacy protection algorithms also utilizes this property of JPEG such that we



could produce a special protected secure JPEG file that is still backward compatible with standard JPEG. In addition to JFIF and Exif, the remaining APPn markers are used for other applications, e.g. APP2 for tagging International Color Consortium (ICC) profile and APP14 as Adobe tag to store image encoding information for DCT filters. In the implementation of our Secure JPEG protection algorithms, we use the APP11 markers to signal the information about the security metadata of protected images.

## 3.2 Secure JPEG Framework

Before introducing our algorithms for protecting image privacy, in this section, we recap the Secure JPEG, an open and flexible framework to secure JPEG images, initially proposed by Dufaux and Ebrahimi in [58].

Secure JPEG framework aims to enable various security options for JPEG images offering similar features as those in JPSEC [62, 63], the framework for security solutions for JPEG2000 images. In other words, Secure JPEG acts as an extension of JPEG and accomplishes for JPEG what JPSEC is enabling for JPEG 2000. The goal of Secure JPEG is to allow the efficient integration and use of security tools enabling a variety of security services in JPEG image. The framework is designed in such a way that it does not interfere with baseline JPEG decoders unaware of such an extension, namely, backward-compatible with JPEG. To signal the information about the security tools used to protect the image, a new marker segment is introduced in Secure JPEG containing information similar to the JPSEC SEC marker segment. This marker segment is present in the Frame Header of the JPEG code-stream. The syntax used can be either similar to that defined by JPSEC, or defined in special structures suitable for specific applications. Similar to JPSEC, Secure JPEG enables the use of various tools supporting a number of security services for JPEG image, including but not limited to:

- **Confidentiality:** Transformation of image data (and/or the associated metadata) into an encrypted/ciphered form such that original information is concealed;
- **Integrity Verification:** Detection of manipulations applied to image data (and/or the associated metadata) to verify its integrity;
- **Source Authentication:** Verification of the identity of a user or party that generated the image content;
- **Conditional Access:** Mechanism to grant or restrict access to image data, including the entire image, parts of image data, or just a low resolution image without being able to visualize a higher resolution;
- **Registered Content Identification:** Registration of a JPEG image with a Registration Authority.



# 4 Secure JPEG Scrambling

Three use cases of the Secure JPEG framework addressing integrity verification, encryption and scrambling are briefly described in our previous study [58]. Particularly, the scrambling method operates by randomly changing the signs of quantized DCT coefficients of a JPEG image. Similar scrambling scheme is also designed in [64] as a tool for protecting privacy in video surveillance systems. In this chapter, we elaborate the scrambling algorithm for securing JPEG image, and evaluate its performance in regard to different aspects of usage in the scenario of social networks and photo sharing. Both objective and subjective experiments were conducted, in comparison with another well-known JPEG-based privacy protection scheme, P3 [32].

The rest of the chapter is structured as follows. Section 4.1 describes in detail the Secure JPEG Scrambling algorithm. Section 4.2 presents the experiments and evaluation results. Finally Section 4.3 summarizes this chapter.

## 4.1 JPEG Scrambling: The Algorithm

The operating principle of JPEG Scrambling is to randomly change the signs of quantized DCT coefficients based on a secret key. The scrambled image is still of the same structure as standard JPEG and is therefore backward compatible with JPEG. With a special decoder or transcoder that supports the proposed scrambling scheme and a correct secret key, the scrambled image can be descrambled to its original form. Such a scrambling scheme can be integrated in either a JPEG encoding/decoding procedure or JPEG transcoding. Figure 4.1 illustrates the two typical types of scrambling protection and recovery processes.

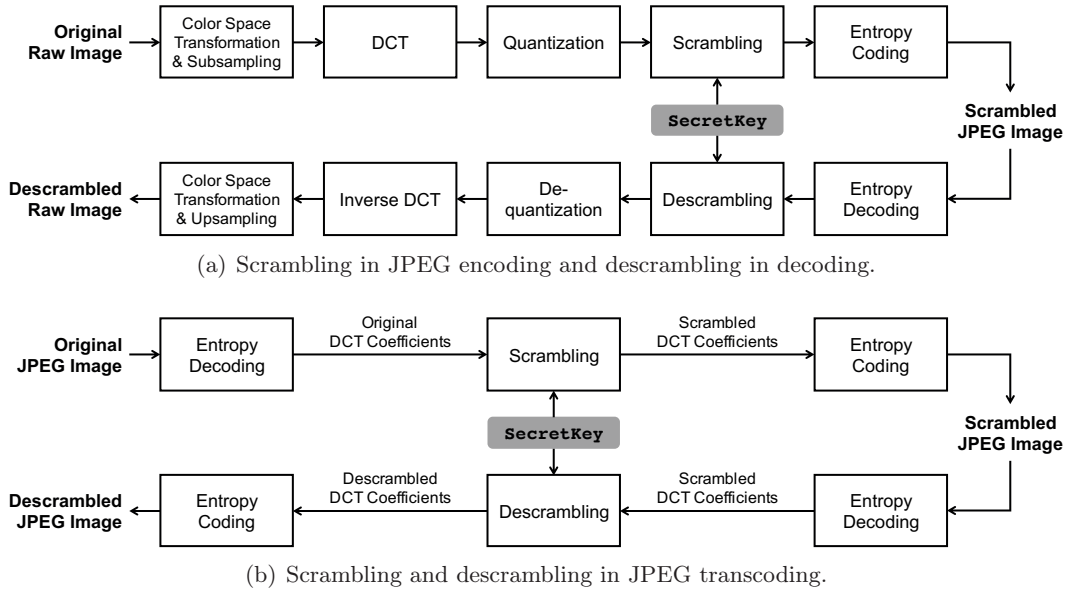


Figure 4.1 – Workflow of JPEG Scrambling protection and recovery in (a) JPEG encoding/decoding or (b) JPEG Transcoding.

### 4.1.1 The Scrambling Protection

The scrambling protection process is carried out throughout either an entire image or arbitrary image regions with MCU blocks as the minimum units. Therefore, our algorithm is also called *selective* scrambling. To define the image regions to protect, we introduce the *Mask Matrix*, noted as  $\mathbf{M}$ , which specifies the shape, size and locations of regions of interest (ROIs) to be scrambled. The mask matrix is a binary-valued 2D matrix, each element of which points to each MCU block of the entire image. In a mask matrix, elements 1 indicate MCU blocks to be changed and 0 for unchanged blocks. An MCU block in JPEG is usually of  $16 \times 16$  pixels for 4:2:0 chroma subsampling and therefore our mask matrix is of size  $\lceil W/16 \rceil \times \lceil H/16 \rceil$  pixels, where  $W$  and  $H$  indicate the width and height of the image and  $\lceil \cdot \rceil$  indicates the ceiling function.

We use a *strength factor*  $l$  to specify the scope of DCT coefficients to scramble, such that different levels of visual obfuscation can be achieved. We define four levels of scrambling strengths ( $l \in \{L, M, H, UH\}$ ) described as follows:

- **Low level (L):** Scramble only the AC coefficients of all three YCbCr components.
- **Medium level (M):** Scramble both the DC and AC coefficients of only the Y component.
- **High level (H):** Scramble both the DC and AC coefficients of all YCbCr components.

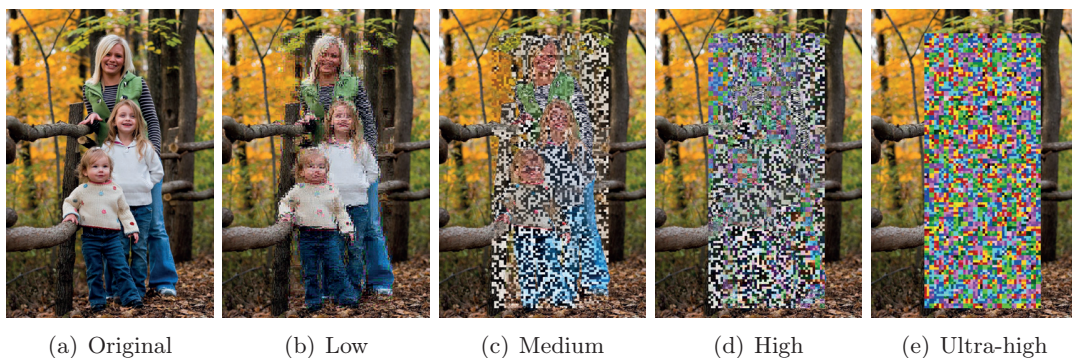


Figure 4.2 – Example image scrambled in different strength levels.

- **Ultra-High level (UH):** Scramble both the DC and AC coefficients of all YCbCr components followed by applying a XOR cipher on the DC coefficients of all three components.

Figure 4.2 shows an example image scrambled in different strength levels. According to the mask matrix  $\mathbf{M}$  and above definitions of scrambling strength levels, a function  $\mathcal{F}()$  is applied on the original JPEG image  $I_O$  to generate a vector of DCT coefficients (noted as  $\mathbf{COEF}$ ) that need to be actually scrambled:

$$\mathbf{COEF} = \mathcal{F}(I_O, \mathbf{M}, l). \quad (4.1)$$

Based on a *Secret Key*, we generate a binary key stream  $\mathbf{KA}$ , of the same length as  $\mathbf{COEF}$ . Then DCT coefficients in  $\mathbf{COEF}$  are scrambled according to the key stream by the following formula:

$$\mathbf{COEF}^* = \mathbf{COEF} \odot (\mathbf{KA} \times 2 - 1), \quad (4.2)$$

where  $\odot$  denotes elementwise multiplication and  $\mathbf{COEF}^*$  denotes the vector of scrambled DCT coefficients. If  $l = \text{UH}$  (Ultra-high level specified), the DC coefficients in  $\mathbf{COEF}^*$  are further encrypted with a new key stream  $\mathbf{KA}'$  by a bitwise XOR operation:

$$\mathbf{COEF}_{\text{DC}}^* = \mathbf{COEF}_{\text{DC}}^* \oplus \mathbf{KA}', \quad (4.3)$$

where  $\mathbf{COEF}_{\text{DC}}^*$  is a subset of  $\mathbf{COEF}^*$  containing only DC coefficients and  $\oplus$  denotes the bitwise XOR operator. Afterwards, the modified DCT coefficients within the ROIs along with the original coefficients out of the ROIs are entropy coded.

As the last but indispensable step, a set of metadata about the scrambled image is inserted in a JPEG application marker (APP11 in our current design) in the scrambled JPEG file. The metadata includes the strength factor  $l$  and the mask matrix data. Since

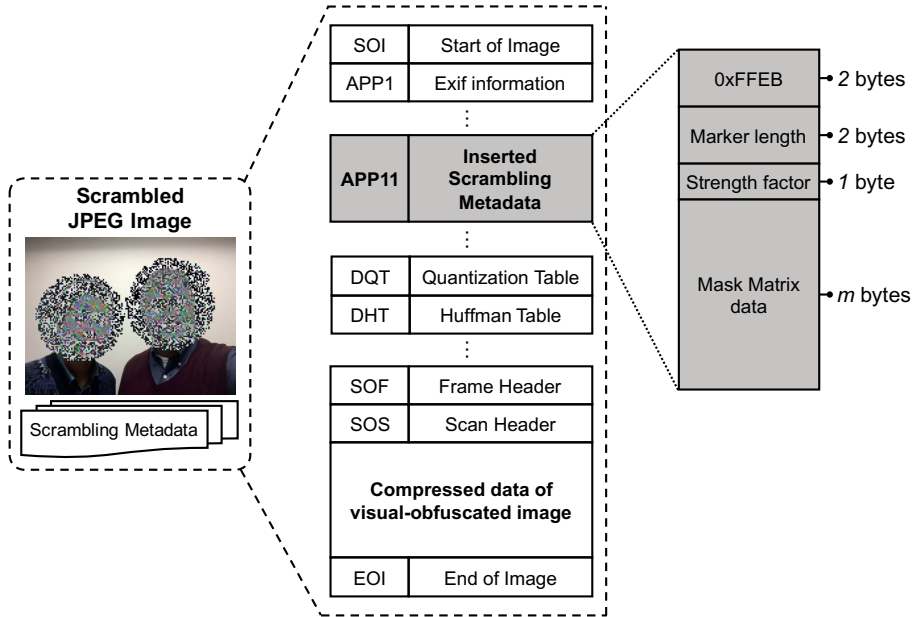


Figure 4.3 – The syntax of scrambled JPEG image file.

the mask matrix contains only 0 and 1, we simply encode it as a binary sequence, with each bit representing an element of the matrix. In practice, the mask matrix can be further compressed as it is sparse. Figure 4.3 shows the syntax of a scrambled JPEG file, which follows the same structure of standard JPEG but contains an extra APP11 marker signaling the metadata about the scrambling protection.

### 4.1.2 The Descrambling Recovery

The descrambling process aims to recover the original image from the scrambled image, with the same secret key used in the scrambling protection. Given the scrambling procedure described in Section 4.1.1, the descrambling process basically reverses the operations performed in scrambling. First, a special JPEG decoder/transcoder is needed to extract the strength factor and mask matrix data from the APP11 markers of the scrambled image. While reading the entropy-decoded JPEG data, the vector **COEF\*** holding the scrambled DCT coefficients is formed based on the mask matrix specifying the positions of MCU blocks and the strength factor specifying the scope of scrambled DCT coefficients. The same binary key stream **KA** (and **KA'** is case of Ultra-high level scrambling) is generated if a correct secret key is provided. Then, the descrambled DCT coefficients **COEF** are generated by the given operation:

$$\mathbf{COEF} = \mathbf{COEF}^* \odot (\mathbf{KA} \times 2 - 1). \tag{4.4}$$

If  $l = \text{UH}$  (Ultra-high level specified), the DC coefficients in  $\mathbf{COEF}^*$  are first decrypted using the bitwise XOR operation before the above descrambling operation:

$$\mathbf{COEF}_{\text{DC}}^* = \mathbf{COEF}_{\text{DC}}^* \oplus \mathbf{KA}', \quad (4.5)$$

Finally, the descrambled DCT coefficients  $\mathbf{COEF}$  of the protected ROIs along with other original coefficients are either encoded to form the descrambled JPEG image or fed into a JPEG decoder for display.

## 4.2 Performance Evaluation

In this section, the performance evaluation of the JPEG Scrambling is reported, in comparison with another well-known JPEG-based privacy protection method, P3 [32]. Such a scrambling scheme may cause expansion to image file size due to the random modifications applied to DCT coefficients and the inserted metadata. In practice, such expansion is expected to be as small as possible to minimize the storage overhead and transmission burden. Unlike encryption, such an image scrambling may still present certain amount of original image visual information. Therefore, we first evaluate and compare the storage overhead created by the two protection methods, and then investigate how well the two methods are able to preserve privacy against different attacking scenarios.

### 4.2.1 Storage Overhead

**Evaluation Metric** First of all, we give the definitions of storage overhead (noted as  $O$ ) for JPEG Scrambling and P3 as follows respectively:

$$O_{\text{Scrambling}} = \frac{S(I_S) - S(I_O)}{S(I_O)} \quad \text{and} \quad O_{P3} = \frac{S(I_{P3}^{\text{Pub}}) + S(I_{P3}^{\text{Sec}}) - S(I_O)}{S(I_O)}, \quad (4.6)$$

where  $S(I)$  indicates the file size of an image  $I$  and  $I_O$ ,  $I_S$ ,  $I_{P3}^{\text{Pub}}$  and  $I_{P3}^{\text{Sec}}$  denote the original image, the scrambled image, the public and secret part of P3-protected image respectively. Since P3 protection splits an image in two portions: public part and secret part, separately stored in client- and server-side, we need to take into account both portions when computing the storage overhead.

**Dataset** We evaluate the storage overhead based on three publicly available image datasets, as representatives of different types of images:

- The **USC-SIPI** image database<sup>1</sup> [65], which contains 215 raw images in Tagged Image File Format (TIFF) with various sizes such as  $256 \times 256$ ,  $512 \times 512$ , or  $1024 \times 1024$

---

<sup>1</sup><http://sipi.usc.edu/database/>

pixels. There are 53 images in color and the rest in grayscale. From the dataset, we removed 6 images that have special spatial patterns such as chessboard-like blocks, to minimize the influence of ROI selection on storage overhead<sup>2</sup>. We convert all the TIFF images to JPEG with a quality factor of 85. The file size of most resulted JPEG images ranges between 7 and 398 KB. This dataset is a representative set of small-size standard images containing various image content.

- The People in Photo Album (**PIPA**) dataset<sup>3</sup> [66], which consists of over 60000 images (JPEG format) of more than 2000 individuals collected from public Flickr photo albums. Each image in this dataset contains one or more people in image content. From this dataset, we randomly selected 1500 images of the same size  $1204 \times 768$  pixels<sup>4</sup>. The file sizes of the 1500 images are in range of 105 KB and 945 KB. This is a representative collection of small to medium size internet images with people as the major content.
- The **INRIA** Holiday image dataset<sup>5</sup> [67], which contains 1491 full color images (in JPEG format) from vacation scenes (e.g. mountain, a river, a small town and other interesting topographies.). It has a greater diversity than the USC-SIPI and PIPA datasets in terms of image content, texture and resolutions. Unlike USC-SIPI and PIPA, images in this dataset were captured from digital cameras directly. The image file sizes of this dataset fall in range from 82 KB to 6.35 MB. This is a representative dataset of medium to large size photographic images.

**Experiments and Analysis** For each image from the three datasets, we manually created 10 mask matrices representing different ROIs with increasing size (10% to 100%) relative to the entire image area. We then applied JPEG Scrambling (in four strength levels) on the 10 ROIs of each image respectively. Meanwhile, we applied P3 using four different threshold values ( $t = 1, 5, 10$  and  $20$ ) on each entire image. Note that the original P3 algorithm does not directly supports partial protection. The storage overhead for each protected image (JPEG Scrambling and P3) is computed and the results (mean and 95% confidence interval) for the three datasets are shown in Figure 4.4. From the results, one observes a near linear relation between the storage overhead and the relative size of protection ROI, for any level of JPEG Scrambling. The slopes for different levels of scrambling are different: the growth rate is higher if higher level of scrambling is applied. For Low-, Medium- and High-level scrambling, the overhead is extremely low:  $< 6\%$  for USC-SIPI,  $< 8\%$  for PIPA and  $< 3\%$  for INRIA. The overhead for Ultra-high level scrambling, though about double the overhead of High-level scrambling, is still acceptable:

---

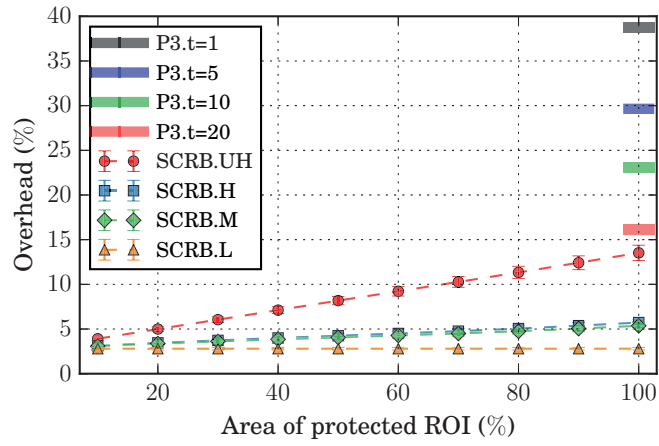
<sup>2</sup>The final subset of 209 images used in the experiment is available at <http://grebvm2.epfl.ch/lin/thesis/dataset/USC-SIPI-subset-209.zip>.

<sup>3</sup><https://people.eecs.berkeley.edu/~nzhang/piper.html>

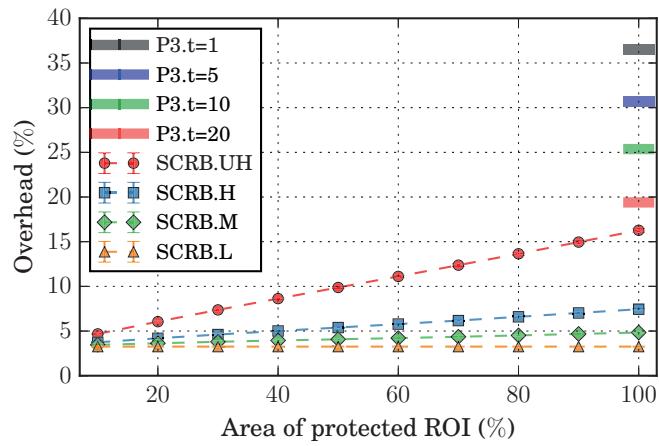
<sup>4</sup>The final subset of 1500 images used in this experiment is available at <http://grebvm2.epfl.ch/lin/thesis/dataset/PIPA-subset-1500.zip>

<sup>5</sup><http://lear.inrialpes.fr/people/jegou/data.php#holidays>

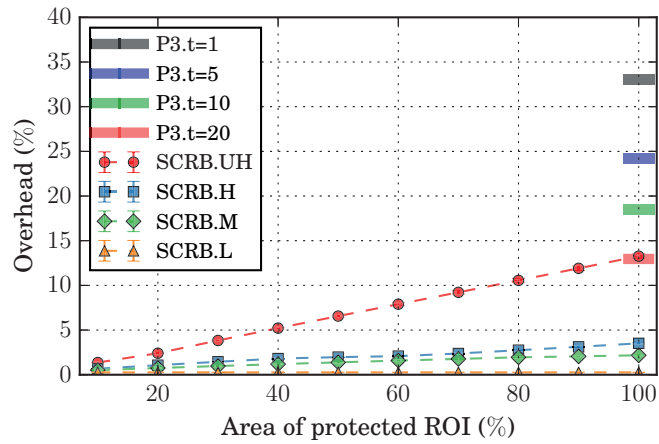




(a) USC SIPI dataset



(b) PIPA dataset



(c) INRIA Holidays dataset

Figure 4.4 – Storage overhead of JPEG Scrambling on different image datasets.

for all the three datasets, the maximum overhead for entire image scrambling (ROI size = 100%) is not higher than 17%. Compared to JPEG Scrambling, the overhead of P3 (for threshold between 1 and 20) is much higher. Although increasing the threshold reduces P3's overhead, higher threshold values make P3 public images preserve more information of the original image, therefore revealing more visual privacy. This will be proved in Section 4.2.2. The mean overhead when images are entirely scrambled in any level of strength is always lower than or equal to that of P3 in threshold of 20, not to mention partial image region scrambling. From this experiment, JPEG Scrambling outperforms P3 in terms of storage overhead.

### 4.2.2 Privacy Protection Capability

We also evaluated the privacy protection capability of the two algorithms (JPEG Scrambling and P3) in two typical scenarios of recognition attacks that may compromise users privacy: (i) face detection/recognition and (ii) number plate recognition. The two types of attacks aim at identifying different visual information, namely the human face and text (number and letter). We conducted both objective and subjective experiments, to investigate the performance of the protection methods against different “attackers”, i.e. machine and real human.

#### Objective Privacy Evaluation

**Face Detection** We first conducted a face detection experiment using the Caltech face dataset<sup>6</sup>, which contains 450 frontal color face images of 27 individuals depicted in different circumstances (illumination, background, facial expressions, etc.). We first applied JPEG Scrambling (four strength levels) and P3 ( $t = 1, 5, 10$  and  $20$ ) on each image, and then applied the Haar face detector [68] from the OpenCV library<sup>7</sup> on each original image and its protected variants. For P3, the face detection is performed on the public image. The detection rates (proportion of correctly detected faces) for original and protected images are shown in Table 4.1.

From the results, one observes that automatic face detection still performs well in Low-level scrambled images, where more than 90% faces were successfully detected. When Medium- or High-level scrambled was applied, the detection rate is greatly reduced to 2.99% and 0.85% respectively. With Ultra-high level scrambling applied, not a single face could be detected. Compared to JPEG Scrambling, P3 in all four threshold values always provides stronger protection such that no any face could be detected.

---

<sup>6</sup>[http://www.vision.caltech.edu/Image\\_Datasets/faces/](http://www.vision.caltech.edu/Image_Datasets/faces/)

<sup>7</sup><http://opencv.org/>

Table 4.1 – Results of objective privacy evaluation of JPEG Scrambling and P3: face detection rate, license plate recognition rate and SSIM (mean).

Image type	Face detection	License plate recognition	SSIM
Original	100%	82.5%	1
SCRB.L	94.23%	0.20%	0.497
SCRB.M	2.99%	0	0.160
SCRB.H	0.85%	0	0.161
SCRB.UH	0	0	0.093
P3.t=20	0	1.39%	0.682
P3.t=10	0	0.40%	0.633
P3.t=5	0	0.40%	0.589
P3.t=1	0	0.20%	0.452

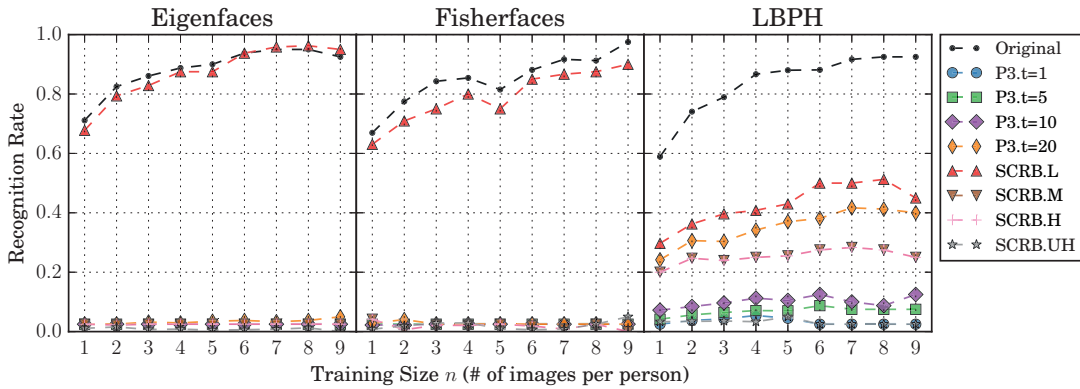


Figure 4.5 – Face recognition results obtained on the original and different protected images using three different recognition methods: Eigenfaces, Fisherfaces and LBPH.

**Face Recognition** We then evaluated face recognition using the AT&T face image dataset<sup>8</sup>, which contains 40 individuals' frontal face images (10 images for each identity). We implemented three classical face recognition methods, namely the Eigenfaces [69], the Fisherfaces [70], and the Local Binary Patterns Histograms (LBPH) [71], using the OpenCV library<sup>9</sup>. For each identity, we selected  $n$  of his/her images (in original form without protection) as training set, leaving the rest  $10 - n$  images as the evaluation set. Each image in the evaluation set were protected by the two methods with the same settings as the previous experiment. Each recognition classifier was trained using the  $n$  training images of each identity, and evaluated on different variants (original images or protected images with different methods) of the evaluation set. The overall recognition rate (proportion of correctly identified faces) of all the 40 identities corresponding to different values of  $n$  for different recognition algorithms is shown in Figure 4.5.

<sup>8</sup><http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>

<sup>9</sup><http://docs.opencv.org/2.4/modules/contrib/doc/facerec/>

First of all, one observes that the face recognition rate for original unprotected evaluation images generally increases as the training size increases, regardless of recognition method. When 7 or more images (per person) were used for training, the recognition rate for original images reaches the scale higher than 0.9. For methods Eigenfaces and Fisherfaces, Low-level JPEG Scrambling does not effectively reduce the recognition rate. If higher-level scrambling has been applied (M, H or UH), the recognition rate is significantly reduced to nearly 0. For P3, any of the four parameters almost deactivates face recognition by methods Eigenfaces and Fisherfaces. Compared to Eigenfaces and Fisherfaces, LBPH is more robust to most of the visual obfuscations. For Low- to High-level scrambling and P3 ( $t=20$ ), the recognition rate is lower than half of the rate obtained on original images, though still considerably higher than that of a random guess. When stronger obfuscation is applied (Ultra-high level scrambling or P3 with  $t \leq 10$ ), the recognition rate becomes as low as random guess.

**License Plate Recognition** To evaluate the performance of license plate recognition on different types of protected images, we employed a license plate image dataset from <http://www.zemris.fer.hr/projects/LicensePlates/english/results.shtml>. The dataset contains 503 images of the rear views of various vehicles (cars, trucks, busses), taken from an OLYMPUS C-2040 ZOOM digital camera under various lighting conditions. The two protection methods with the same parameters as in the previous experiments were applied on each image. We then used the open source license plate recognition library OpenALPR<sup>10</sup> to identify the license plate in original and protected images. The recognition rate (the proportion of correctly identified plates) for different protection setups is shown in Table 4.1.

As is shown, the recognition rate for original license plate images is about 82.5%. With Low-level JPEG Scrambling applied, only one license plate (about 0.2%) was recognized. By applying higher level JPEG Scrambling, not a single license plate could be correctly identified. The performance of P3 is comparable with JPEG Scrambling, but slightly worse. The recognition rates are 1.39%, 0.40%, 0.40% and 0.20% for threshold 20, 10, 5 and 1 respectively. Again, this experiment indicates that JPEG Scrambling provides comparable level of protection as P3 in making number plate unintelligible.

### Subjective Privacy Evaluation

Similarly, two sets of subjective experiments were conducted based on online crowdsourcing to evaluate how real humans perform in different recognition tasks, i.e. face recognition and license plate recognition from protected images.

---

<sup>10</sup><http://www.openalpr.com/>

**Face Recognition** For the subjective experiment on face recognition, we employed the color Face Recognition Technology (FERET) dataset<sup>11</sup> [72], from which we selected 9 male identities (4 white, 3 black and 2 Asian). In the experiment, we recruited online subjects through Amazon Mechanical Turk (AMT)<sup>12</sup>, and asked subjects to identify the person in protected images from the 9 identities. We selected 6 face images for each identity: 2 regular frontal, 2 left view and 2 right view images. For each identity, we made three of his images (one frontal, one left view and one right view) public and unprotected as the training set, with the rest three as evaluation set. Each image in the evaluation set was protected by JPEG Scrambling and P3 with different parameters as before. Finally, 216 different protected images ( $3 \times 9 \times 8$ ) were generated. In each Human Intelligence Task (HIT) on AMT, the training images of all the 9 identities are firstly presented and made always available during the experiment session for subjects to review. Then an evaluation image protected by a certain method is presented with a question asking the subject to identify the person in image. If the subject has no any clue about the identity of the protected person, he/she could choose the option “I really don’t know”. In addition to the evaluation set, we also included the original images from the training set in evaluation, serving as “honeypot” to help us remove sloppy subjects. We asked 20 subjects to vote on each image. A screenshot of an HIT on AMT is shown in Figure A.1 in Appendix A.

Finally, 146 valid subjects<sup>13</sup> completed the experiment each voting on 33.3 images in average. The proportion of correct, incorrect and “I don’t know” answers corresponding to each protection is shown in Figure 4.6(a). Here, one observes that the face recognition rate (proportion of correct answers) for Low-level JPEG Scrambling is still high, close to 100%. This is similar as the automatic face recognition results obtain from Eigenfaces and Fisherfaces. When Medium- or High-level JPEG Scrambling was applied, the recognition rate decreased to about 60% and 40% respectively. Only when Ultra-high level scrambling applied, almost no face could be correctly identified and most people selected the answer “I don’t know”. As for P3, threshold values of 20 and 10 provide similar degree of privacy protection as Medium- and High-level JPEG Scrambling. Similar to JPEG Scrambling, only with the strongest protection ( $t = 1$ ) applied, no face can be correctly recognized.

**License Plate Recognition** A subjective experiment on number plate recognition was conducted in a similar way. From the license plate image dataset used in the objective experiment, we selected a subset consisting 28 example license plate images, shown in the web page<sup>14</sup>. Every license plate image was protected by the two methods each with the corresponding four parameters. In each HIT on AMT, a license plate image (either

<sup>11</sup><https://www.nist.gov/itl/iad/image-group/color-feret-database>

<sup>12</sup><https://www.mturk.com/>

<sup>13</sup>Subjects who provided wrong answers to “honeypot” images were removed and vacated HITs were republished on AMT until all have been successfully finished.

<sup>14</sup><http://www.zemris.fer.hr/projects/LicensePlates/english/images.html>

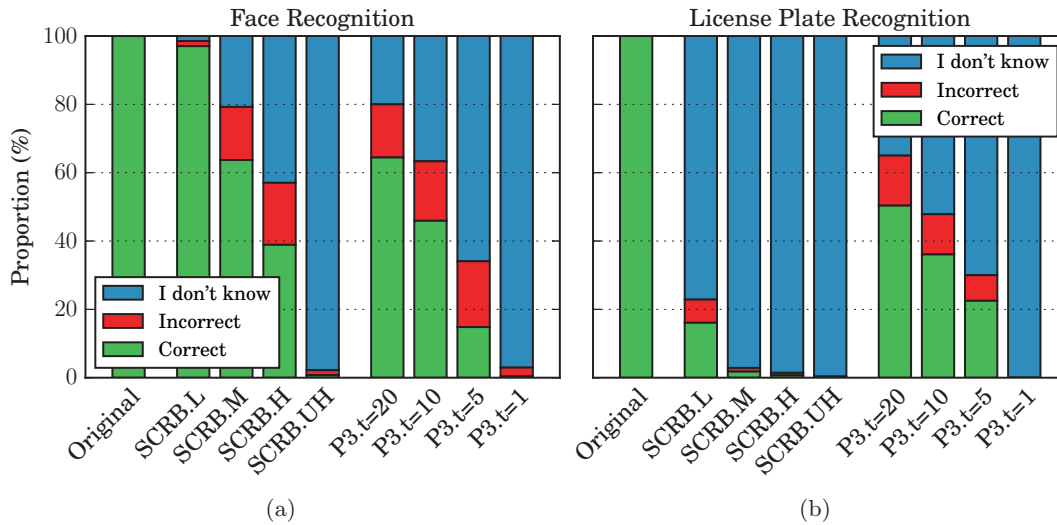


Figure 4.6 – Results of subjective experiment on (a) face recognition and (b) license plate recognition: proportion of “I don’t know”, incorrect and correct answers for original and protected images.

original or protected form) is presented to the subject, who is asked to identify the license plate and record it in a text field. If the subject cannot get any clue from protected image, he/she can simply put “0” indicating “I don’t know”. A screenshot of an HIT for license plate recognition on AMT is provided in Figure A.2 in Appendix A. Original license plate images were used in evaluation as “honeypot” for detecting outliers. Again, we asked 20 subjects voting on each image, which resulted in 112 subjects each annotating 45 images on average. Sloppy subjects who provided wrong answers to original images were removed.

The recognition results are shown in Figure 4.6(b). As is revealed, the recognition rate in Low-level scrambled license plate images is already very low, below 20%. With higher-level scrambling applied, the recognition rate is further reduced, extremely close to zero. As for P3, with threshold values of 20, 10 and 5, there are still a large number of license plates correctly recognized ( $\sim 50\%$ ,  $\sim 33\%$  and  $\sim 22\%$  respectively). Only with the strongest protection applied ( $t = 1$ ), no license plate can be recognized. The results of this experiment also agree with that of automatic license plate recognition experiment: For both real human and machines, JPEG Scrambling is superior to P3 in protection of text information in image, unless the strongest P3 is applied ( $t=1$ ).

## Analysis and Discussions

The experiment reveals that the two methods perform differently in protecting different types of visual information. For instance, when protecting privacy against face recognition,

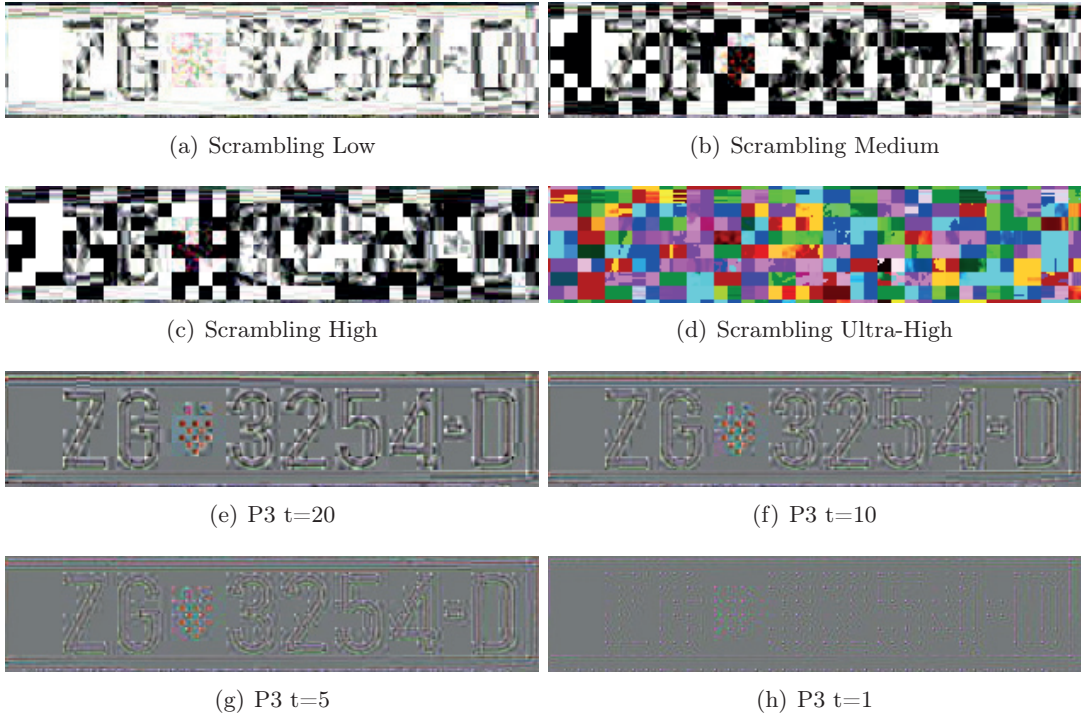


Figure 4.7 – Example license plate protected by JPEG Scrambling and P3 with different parameters.

the performances of JPEG Scrambling in M, H and UH levels are comparable with that of P3 in  $t = 20$ , 10 and 1 respectively. While, for license plate recognition, the performances of JPEG Scrambling in the three levels are significantly better than the corresponding levels of P3. This is due to the different natures of the two methods: JPEG Scrambling is based on permutation of signs of DCT coefficients, which interfere with image information in all frequencies. While, P3 uses a threshold to transfer partial information from each DCT coefficient to a secret image, preserving minimal amount of information about the original image in the public image. If the threshold in P3 is not small enough, significant high-frequency information might be disclosed from the public image, as many high-frequency AC coefficients have small values. These features include sharp contours or edges, such as numbers or letters in license plate. An example license plate image obfuscated by the two methods with different parameters is shown in Figure 4.7. In scrambled images, the license plate can be hardly observed even for Low-level scrambling. However, one can easily read the license plate in P3-protected images for  $t \in [5, 10, 20]$ . To quantify how much information about the original image is revealed from an obfuscated image, we use another metric, namely the structural similarity index (SSIM) [73], to measure the similarity between the original and the protected image. We computed and the SSIM for each protected image from the Caltech face dataset compared to its original image. The mean values of SSIM corresponding to different protections are listed in

Table 4.1. As is shown, images protected by P3 (the public part) still reveal somewhat similarity compared to the original images. For the threshold of 20, an average SSIM of 0.682 is obtained. Even with the lowest threshold of 1, an average SSIM of 0.452 is also resulted. Compared to P3, the SSIM scores of scrambled images are much lower: 0.497 for Low-level scrambled images, about 0.16 for Medium- or High-level scrambled images, and merely 0.093 for Ultra-high scrambled images. Although SSIM of protected image is not directly related to its capability of privacy preservation, it reveals a significant difference in the visual appearance between the two different approaches.

### 4.3 Conclusion

This chapter elaborates JPEG Scrambling, a lightweight encryption for securing JPEG image, initially introduced in [64] as a tool for protecting privacy in video surveillance. JPEG Scrambling randomly changes the signs of quantized DCT coefficients of a JPEG image, based on a secret key. It is a selective approach in the sense that arbitrary ROIs can be defined and different levels of strength can be achieved. Both objective and subjective experiments were conducted to evaluate the performance of the proposed method with regard to its storage overhead and privacy preservation capability, in comparison with another JPEG-based privacy protection scheme, P3 [32]. Experimental results indicate that JPEG Scrambling provides similar performance as P3 in preventing face recognition from both machines and real human. In addition, the scrambling approach shows its advantages over P3 in protecting text information in image, due to different nature of visual information permutation. The privacy protection capabilities of the two methods both vary depending on the selected strength parameter, which in turn influences the level of storage overhead. For both JPEG Scrambling and P3, storage overhead increases as the protection strength increases. But the overhead of JPEG Scrambling is always lower than P3 even if the strongest scrambling is applied on the entire image. Using the proposed JPEG Scrambling, we could always achieve a stronger protection but less storage overhead than P3; in this respect, JPEG Scrambling outperforms P3. A significant drawback of JPEG Scrambling is that scrambled image is not robust to most lossy image transformations, meaning that the reconstruction becomes impossible if a scrambled image has been modified. This is because the applied transformation may completely reorder the signs of DCT coefficients. Also, similar to any other encryption-based approach, JPEG scrambling can only generates high distorted visual effect, which may not be expected when applied in protection of images in social media.



## 5 Secure JPEG Transmorphing

Most approaches to protect image visual privacy stay in the stage of encrypting or permuting image data, including the proposed JPEG Scrambling in the last chapter. From data security point of view, an encryption-based scheme can well preserve privacy in a secure and reversible manner. However, simply encrypting an entire image results in either an unreadable image or highly distorted visual effect, which may significantly affect the usability of photo sharing and may not be in users best interest from both usage and perception perspectives. In many cases, people hope to share their photos online to public while partially hiding specific image regions in a simple and pleasant way, e.g. creating an anonymous face with a cartoon smiley, or blurring and inpainting a sensitive area. Usually, those interesting manipulations cannot be reversed directly. Inspired by these facts, we explore the design of visual protection method that can satisfy all the characteristics raised in the introduction of the thesis (Chapter 1), including not only basic requirements such as security, reversibility, robustness, backward-compatibility, but also advanced features such as personalization and pleasantness.

In this chapter, we present secure JPEG Transmorphing, an image privacy protection algorithm, or framework, that meets all desired features outlined in Chapter 1. Within Secure JPEG Transmorphing, almost any type of regional obfuscation can be applied, such as masking, blurring, pixelation, inpainting, warping, etc. More importantly, the original image can be reconstructed with near lossless quality, even if the protected image has been manipulated. A set of experiments were conducted to evaluate the performance of the proposed approach with respect to its storage overhead, reconstruction quality, privacy protection capability and subjective pleasantness.

The rest of this chapter is structured as follows. Section 5.1 describes in detail the proposed JPEG Transmorphing. Section 5.2 reports the evaluation experiments and results analysis. Section 5.3 outlines some discussions and Section 5.4 summarizes this chapter.

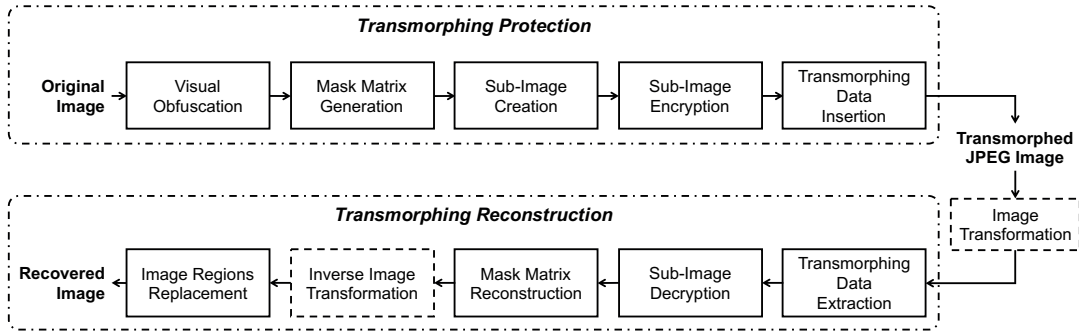


Figure 5.1 – Overview of JPEG Transmorphing: the protection and reconstruction procedures.

## 5.1 JPEG Transmorphing: The Algorithm

The working principle of Secure JPEG Transmorphing is to utilize the JPEG application segments markers (APPn) in file header to secretly preserve partial original image information, while encoding the JPEG image in a visually protected form. The visual information of the original image can be protected by any type of regional manipulation, such as masking, blurring, pixelation, inpainting, warping, and so on. The protected image, or called the Transmorphed image, of the same syntax as standard JPEG, is therefore backwards compatible with JPEG. With a dedicated JPEG transcoder or decoder that supports JPEG Transmorphing, the original image can be recovered by replacing the obfuscated regions in the protected image with the corresponding original regions extracted from APPn markers. The workflow of Secure JPEG Transmorphing, comprising two procedures: protection and reconstruction, is illustrated in Figure 5.1. The two procedures are then described in detail as follows.

### 5.1.1 Transmorphing Protection

The protection procedure of secure JPEG Transmorphing consists of three steps: (i) mask matrix generation, (ii) sub-image construction and (iii) Transmorphing data insertion, each presented in the following algorithm blocks. To let reader better understand the following, an illustration of Transmorphing protection is given in Figure 5.2.

#### Mask Matrix Generation

First of all, assume a user attempts to apply an obfuscation on certain regions of interest (ROIs) of an image, such as masking the two faces with cartoon stickers, shown in Figure 5.2. Once finishing the masking operation, a binary-valued 2D matrix is generated to indicate the shape, size and position of the protected ROIs. As JPEG is coded with respect to MCU block, which is composed of several DCT blocks, we let each element

## 5.1. JPEG Transmorphism: The Algorithm

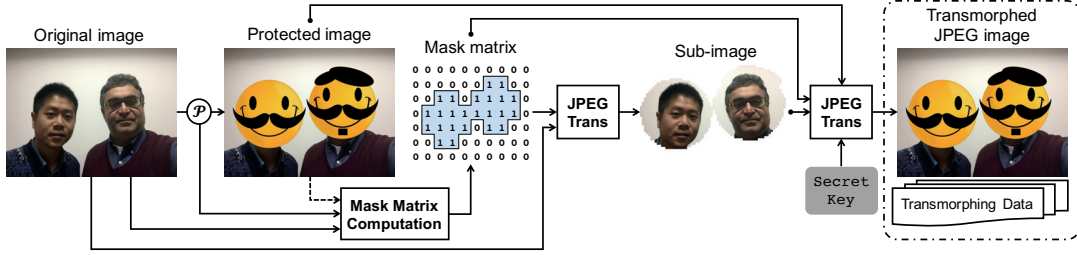


Figure 5.2 – Illustration of a protection procedure of JPEG Transmorphism.

of the matrix point to one MCU block of the image, with elements one to protected blocks and zero to unprotected blocks. As also introduced in Section 4.1.1, this matrix is named *Mask Matrix*, noted as  $\mathbf{M}$ , which holds the essential geometric information about the protected image regions, indispensable for the future reconstruction. Depending on applications, the mask matrix can be generated either from the geometric information of users actions (e.g. coordinates of finger touch on a mobile phone), or by comparing the original and obfuscated images. The step of generating mask matrix is presented in Algorithm 1.

---

### Algorithm 1 $\text{GenerateMaskMatrix}(I_O, \mathcal{P})$

---

```

/* The function to obfuscate partial image ROIs and generate the mask matrix. The
inputs consist of the original image  $I_O$  and an obfuscation operation defined by  $\mathcal{P}$ . */
1:  $I_P \leftarrow \text{VisualObfuscation}(I_O, \mathcal{P})$ 
2: if  $\mathcal{P}$  is known and well defined then
3:    $\mathbf{M} \leftarrow \text{ComputeMaskMatrix}(I_O, \mathcal{P})$     # Compute the mask matrix based on  $\mathcal{P}$ 
4: else
5:    $I_O^Y \leftarrow \text{RGBtoYUV}(I_O)[Y]$                 # Y (luminance) channel of  $I_O$ 
6:    $I_P^Y \leftarrow \text{RGBtoYUV}(I_P)[Y]$                 # Y (luminance) channel of  $I_P$ 
7:    $I_\Delta^Y = |I_O^Y - I_P^Y|$ 
8:    $I_\Delta^B = (I_\Delta^Y > t)$ 
9:    $\mathbf{M} \leftarrow \text{DownSample}(I_\Delta^B, 16)$  # Downsample  $I_\Delta^B$  by factor of 16. If any pixel in an
      MCU block is 1, that block becomes an element 1 in  $\mathbf{M}$ . The mask matrix is round
      up to size of  $(\lceil H/16 \rceil, \lceil W/16 \rceil)$  ( $W$  and  $H$  are width and height of image  $I_O$ ).
10: return  $\mathbf{M}$ 

```

---

### Sub-Image Construction

Based on the mask matrix, a *sub-image* is constructed by transcoding the original JPEG image to a new one while preserving only the DCT coefficients corresponding to ROIs defined by the mask matrix  $\mathbf{M}$ . DCT coefficients outside the ROIs are set to zero. The sub-image is still a JPEG image with the same dimensions but smaller file size as the original image. It contains the information of sensitive part of the original image that the users wants to preserve. This procedure is presented in Algorithm 2.

---

**Algorithm 2** ConstructSubImage( $I_O, \mathbf{M}$ )

---

/\* The function to construct the sub-image  $I_{\text{Sub}}$  from the original image  $I_O$  based on mask matrix  $\mathbf{M}$  \*/

```

1: while Trancoding  $I_O$  to a new JPEG image  $I_{\text{Sub}}$  do           # Loop all MCU blocks
2:    $(i_M, j_M) \leftarrow \text{IndexOfCurrentMCU}()$                  # Index of the current MCU block
3:   if  $\mathbf{M}(i_M, j_M) == 0$  then
4:      $I_{\text{Sub}}.\text{MCUArray}(i_M, j_M) = 0$                        # Set to zero if the MCU is not in the ROIs
5:   else
6:      $I_{\text{Sub}}.\text{MCUArray}(i_M, j_M) = I_O.\text{MCUArray}(i_M, j_M)$  # Copy DCT coefficients
7: return  $I_{\text{Sub}}$ 

```

---

**Transmorphing Data Insertion**

As the last step, the sub-image is secured by a symmetric encryption scheme with a *secret key*, e.g. the Advanced Encryption Standard (AES) [74] or JPEG Scrambling. The security and privacy of the final protected image is ensured by the chosen encryption scheme. Then the bitstream of the encrypted sub-image, the mask matrix, along with a set of metadata, collectively named *Transmorphing data*, is inserted in one or more application segments of the obfuscated JPEG image; in this respect, the obfuscated image serves as a “cover image”. Similar as in JPEG Scrambling, binary elements of the mask matrix is encoded into a bitstream. In practice, the mask matrix can be further compressed as it is sparse. The metadata contains the auxiliary information about the inserted sub-image and mask matrix, such as the data length, the encryption scheme, etc. Since JPEG allows a maximum of 65533 bytes<sup>1</sup> allocated for each marker segment, it is highly probable that the entire sub-image data needs to be separately stored in several APPn segments. In our current implement, APP11 marker is employed for JPEG Transmorphing. The step of inserting Transmorphing data is presented in Algorithm 3 and the syntax of the final Transmorphed image file is illustrated in Figure 5.3.

**Overhead Control**

Inserting additional information in the protected image causes expansion to file size, which will create overhead to storage and transmission. To control such overhead, we designed two mechanisms to reduce the file size of Transmorphed image without sacrificing the quality of reconstructed image. Both mechanisms manipulate the DCT coefficients corresponding to only the obfuscated ROIs in the Transmorphed image (e.g. the image stickers in the “cover image” in Figure 5.3) within a JPEG transcoding process. The manipulations of DCT coefficients can be done (but not limited to) in the following ways:

---

<sup>1</sup>Each JPEG APP marker signals its marker length using two bytes (16 bits), resulting in a maximum of  $(2^{16} - 1) - 2 = 65533$  bytes to record extra information.

## 5.1. JPEG Transmorphing: The Algorithm

---

### Algorithm 3 InsertTransmorphingData( $I_P, \mathbf{M}, I_{\text{Sub}}, \mathcal{C}, K$ )

---

```

/* The procedure to insert the Transmorphing data in obfuscated JPEG image  $I_P$ . Input
parameters include the obfuscated JPEG image  $I_P$ , mask matrix  $\mathbf{M}$ , sub-image  $I_{\text{Sub}}$ , the
chosen encryption scheme  $\mathcal{C}$  and a secret key  $K$ . */
### Insert metadata and mask matrix: ###
1:  $C_{\text{Sub}} \leftarrow \text{Encrypt}(I_{\text{Sub}}, \mathcal{C}, K)$ 
2:  $\text{BS}_{\mathbf{M}} \leftarrow \text{ByteStreamOf}(\mathbf{M})$ 
3:  $\text{MD} \leftarrow [\text{SizeOf}(\text{BS}_{\mathbf{M}}), \text{SizeOf}(C_{\text{Sub}}), \mathcal{C}]$  # Metadata MD
4:  $N_{\text{MD}+\mathbf{M}} \leftarrow \text{SizeOf}(\text{MD}) + \text{SizeOf}(\text{BS}_{\mathbf{M}})$  # Size of the first APP11 segment
5:  $I_P.\text{header.writeAPPnMarker}(\text{"0xFFE8"})$  # Create an APP11 marker
6:  $I_P.\text{header.writeMarkerLength}(N_{\text{MD}+\mathbf{M}})$  # Write the length of the marker
7:  $I_P.\text{header.writeBytes}(\text{MD})$  # Write the bytestream of the metadata
8:  $I_P.\text{header.writeBytes}(\text{BS}_{\mathbf{M}})$  # Write the bytestream of the mask matrix
### Insert sub-image data: ###
1:  $N_{\text{Sub}} \leftarrow \text{SizeOf}(C_{\text{Sub}})$ 
2: if  $N_{\text{Sub}} \leq 65533$  then # Write the sub-image data in one segment
3:    $I_P.\text{header.writeAPPnMarker}(\text{"0xFFE9"})$ 
4:    $I_P.\text{header.writeMarkerLength}(N_{\text{Sub}})$ 
5:    $I_P.\text{header.writeBytes}(C_{\text{Sub}})$ 
6: else # Write the sub-image data in several segments
7:    $N_{\text{Marker}} = \lceil N_{\text{Sub}}/65533 \rceil$  # Number of segments needed for the sub-image data
8:   for  $i \in [1, \dots, N_{\text{Marker}}]$  do
9:      $I_P.\text{header.writeAPPnMarker}(\text{"0xFFE9"})$ 
10:    if  $i \neq N_{\text{Marker}}$  then
11:       $I_P.\text{header.writeMarkerLength}(65533)$ 
12:       $I_P.\text{header.writeBytes}(C_{\text{Sub}}.\text{byteArray}[(i-1) * 65533 : i * 65533])$ 
13:    else
14:       $I_P.\text{header.writeMarkerLength}(N_{\text{Sub}} \bmod 65533)$ 
15:       $I_P.\text{header.writeBytes}(C_{\text{Sub}}.\text{byteArray}[(i-1) * 65533 : \text{end}])$ 
16: return  $I_P$  # Final Transmorphed image  $I_P$ 

```

---

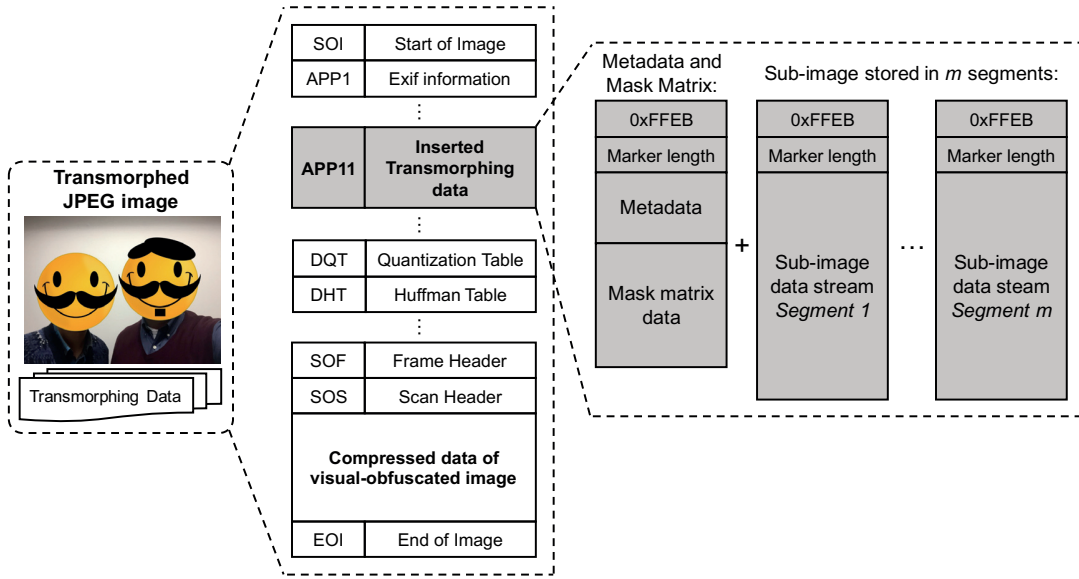


Figure 5.3 – The syntax of Transmorphed JPEG image file.

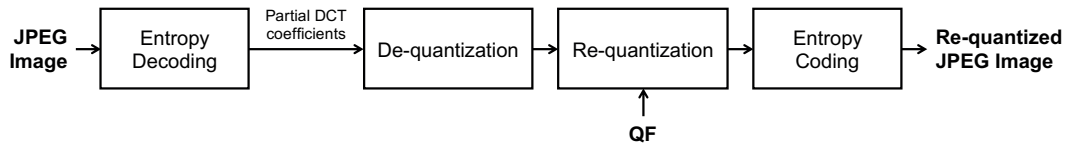


Figure 5.4 – Block diagram of DCT re-quantization for overhead control in JPEG Transmorphing.

**DCT Re-quantization** This approach re-quantizes the DCT coefficients of the obfuscated ROIs. To be more specific, while the transcoding process of JPEG Transmorphing, the DCT coefficients corresponding to obfuscated regions are first de-quantized based on the original quantization table, and then re-quantized using a new quality factor (noted as QF). The quantization steps for re-quantization are computed in the same way as is done in normal JPEG compression, using the QF to scale the standard quantization tables. The re-quantized DCT coefficients of the ROIs along with the original DCT coefficients out of the ROIs are entropy-coded such that the final transmorphed image is generated. However, in the resulted image, only the original quantization tables are kept. If QF is smaller than the original Q factor of the JPEG file, re-quantization greatly reduces the image quality and file size, due to the larger quantization step applied. Even if the applied QF is larger than the original Q factor used in JPEG compression, re-quantizing an existing JPEG image may also decrease file size due to the re-quantization errors, which have been well explained in [75]. We just make sue of such a “side effect” of JPEG re-quantization to decrease the quality of the obfuscated regions in the “cover image”, further reducing the file size of the Transmorphed image. Such a manipulation is illustrated in Figure 5.4.

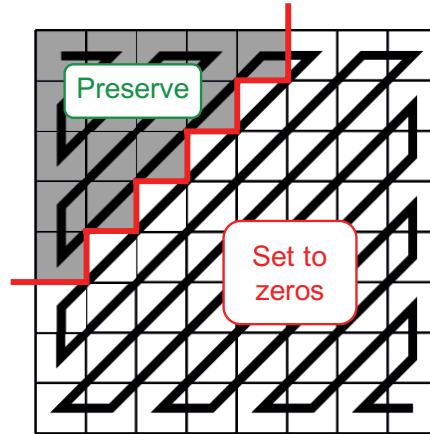


Figure 5.5 – Illustration of DCT coefficients Cut-off with a CF of 5.

**DCT Cut-off** This method simply “cuts off” certain number of high-frequency DCT coefficients by setting them to zero. A Cut-off Factor (CF) is applied to specify the number ( $N$ ) of low-frequency coefficients to preserve in a zig-zag scanning of a DCT block (illustrated in Figure 5.5):

$$N = \frac{(1 + CF) \times CF}{2}. \quad (5.1)$$

An example image Transmorphed with different setups is shown in Figure 5.6. The file size of the Transmorphed image with either DCT Re-quantization or Cut-off is significantly reduced compared to that without overhead control. Although the two approaches may decrease the quality of the protected image, such visual degradation is only observed in the obfuscated regions, e.g. the image stickers, blurred or inpainted image regions, which is less important from a privacy protection point of view. Moreover, since both approaches are applied only on the protected regions in the “cover image” during a JPEG transcoding process, the DCT coefficients of rest regions are kept intact. Therefore, both approaches have no any influence on the quality of the reconstructed image.

### 5.1.2 Transmorphing Reconstruction

The reconstruction procedure aims at recovering the original image from a Transmorphed image, by reversing the above Transmorphing protection operations. Since the inserted mask matrix and sub-image preserves the complete information about the original image corresponding to the protection ROIs, the protected image is robust to most types of image transformations. However, we need to assume that the image transformations do not remove the inserted data in JPEG header, and that the transformation is a known operation that can be re-performed. Depending on whether or not an transformation has been applied on image, the reconstruction process can be done in either the pixel DCT

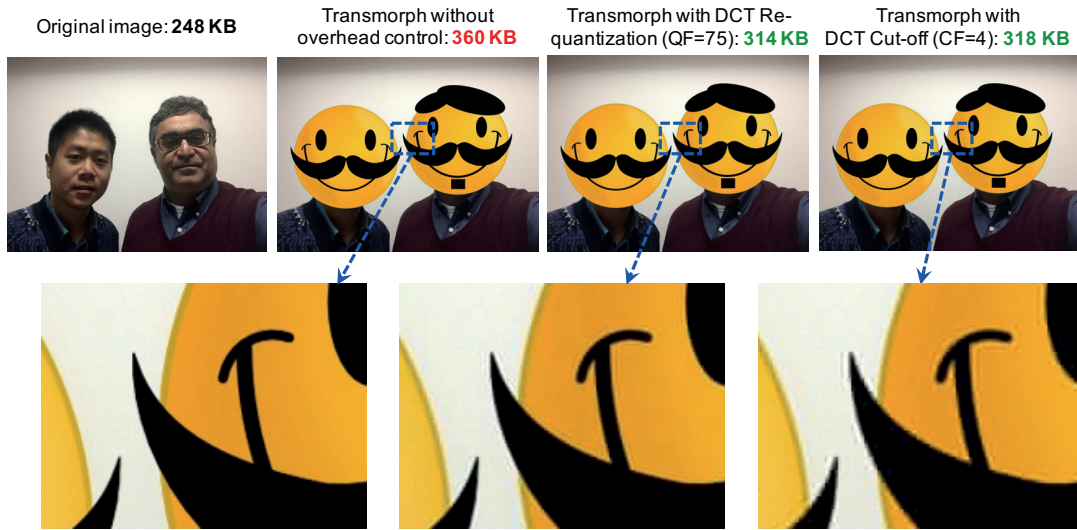


Figure 5.6 – Illustration of an image protected by JPEG Transmorphing without and with overhead controls.

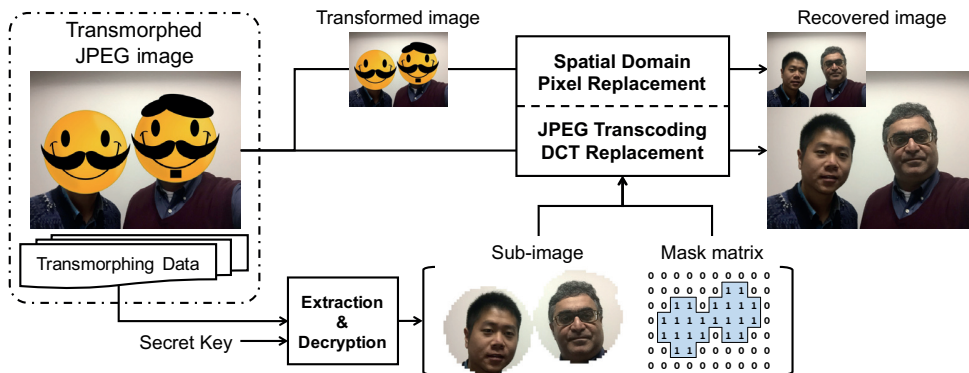


Figure 5.7 – Illustration of a reconstruction procedure of Secure JPEG Transmorphing.

coefficient or pixel domain. The Algorithm 4 presents the reconstruction procedure of JPEG Transmorphing. In addition, we also use a diagram (Figure 5.7) to illustrate the reconstruction procedure of a Transmorphed JPEG image.

## 5.2 Performance Evaluation

In this section, we experimentally evaluate the performance of the proposed JPEG Transmorphing, with regard to the following aspects: (i) storage overhead, (ii) reconstruction quality, (iii) privacy protection capability and (iv) pleasantness.



**Algorithm 4** RecoverTransmorphing( $I_P, \mathcal{T}(\cdot), K$ )

---

```

/* The procedure to recover the original image  $I_{\text{Rec}}$  from a Transmorphed image  $I_P$ , which
might be the outcome of an image transformation  $\mathcal{T}$  applied on the original Transmorphed
image  $I_P^O$  (unavailable).  $K$  is the secret key to decrypt the sub-image. */
1:  $(\mathbf{M}, C_{\text{Sub}}, \mathcal{C}) \leftarrow \text{ExtractTransmorphingData}(I_P)$ 
2:  $\mathbf{M}_O \leftarrow \text{UpSample}(\mathbf{M}, 16)$  # Upsample mask matrix to the size of original image
3:  $I_{\text{Sub}} \leftarrow \text{Decrypt}(C_{\text{Sub}}, \mathcal{C}, K)$ 
4: if  $\mathcal{T}(\cdot)$  was applied, meaning  $I_P = \mathcal{T}(I_P^O)$  then
5:   switch Type of the operation  $\mathcal{T}(\cdot)$  do
6:     case Lossy Geometric Transformation: # Scaling, cropping, warping, etc.
7:        $I'_P \leftarrow \mathcal{T}^{-1}(I_P)$  # Reserve the transformation to get the protected image
of the same geometry as the original image
8:        $I'_P.\text{pixelArray}[\mathbf{M}_O] = I_{\text{Sub}}.\text{pixelArray}[\mathbf{M}_O]$  # Pixels replacement
9:        $I_{\text{Rec}} \leftarrow \mathcal{T}(I'_P)$  # Apply  $\mathcal{T}(\cdot)$  again to get the recovered image  $I_{\text{Rec}}$  of the
same geometry as  $I_P$ 
10:    case Lossy Compression: # E.g., JPEG compression
11:       $I_P.\text{pixelArray}[\mathbf{M}_O] = I_{\text{Sub}}.\text{pixelArray}[\mathbf{M}_O]$ 
12:       $I_{\text{Rec}} = I_P$ 
13:    case Lossless Rotation/Flipping via JPEG Transcoding:
14:       $\mathbf{M}' \leftarrow \mathcal{T}(\mathbf{M})$ 
15:       $I'_{\text{Sub}} \leftarrow \text{JPEGTrans}(I_{\text{Sub}}, \mathcal{T}(\cdot))$ 
16:       $I_P.\text{MCUArray}[\mathbf{M}'] = I'_{\text{Sub}}.\text{MCUArray}[\mathbf{M}']$  # DCT coefficients replacement
17:       $I_{\text{Rec}} = I_P$ 
18:  else # If the protected image is intact without any transformation
19:     $I_P.\text{MCUArray}[\mathbf{M}] = I_{\text{Sub}}.\text{MCUArray}[\mathbf{M}]$ 
20:     $I_{\text{Rec}} = I_P$ 
21:  return  $I_{\text{Rec}}$ 

```

---

**5.2.1 Storage Overhead**

To evaluate the storage overhead, we used the same three datasets and methodology as is used in Section 4.2.1. Namely, we manually created 10 mask matrices for each image, representing different ROIs of increasing size (10% to 100%). We then applied JPEG Transmorphing (without and with overhead control using the two mechanisms) on each of the 10 ROIs in each image. Since JPEG decoding and re-encoding may affect image file size, we directly insert the sub-image into the original JPEG image, instead of actually creating the obfuscated image manipulated in spatial domain to diminish such an impact. This is to diminish the impact of JPEG decoding and re-encoding on file size, and it is equivalent as if we assume the applied image obfuscation does not change the file size. Similarly, the storage overhead for a Transmorphed image is defined as:

$$\mathcal{O}_{\text{Transmorphing}} = \frac{S(I_P) - S(I_O)}{S(I_O)}, \quad (5.2)$$

where  $I_P$  and  $I_O$  refer to the Transmorphed image and original image respectively. We put P3 in comparison and applied P3 on each image with thresholds of 1, 5, 10 and 20 respectively. Finally, for each Transmorphed image, the storage overhead was computed and the results over all images (mean and 95% confidence interval) of each dataset for the two overhead control mechanisms are presented in Figure 5.8 and 5.9 respectively.

Similar to JPEG Scrambling, for different setups of JPEG Transmorphing, a near linear relation between the storage overhead and area of protected ROI is observed. However, the overhead of JPEG Transmorphing is significantly higher than that of JPEG Scrambling, when compared with Figure 4.4 in Section 4.2.1. This is mainly due to the sub-image data inserted in the Transmorphed image. With the proposed overhead control mechanisms, the overhead of JPEG Transmorphing can be greatly reduced. For instance, with DCT Re-quantization (QF=80) or DCT Cut-off (CF=4), the overhead is reduced by more than 40% compared to the case without overhead control. The overhead of P3 (applied on the entire image) is lower than that of JPEG Transmorphing when applied on entire image. Yet, the major purpose of using JPEG Transmorphing is to protect regional image information, instead of obfuscating the entire image. With overhead control applied, the overhead of JPEG Transmorphing can also be effectively reduced. For instance, if we apply JPEG Transmorphing on 40% of the entire image region, using either DCT Re-quantization (QF=80) or DCT Cut-off (CF=4), the overhead is only between 20% and 30% (depending on dataset) and very close to that of P3 in threshold of 5.

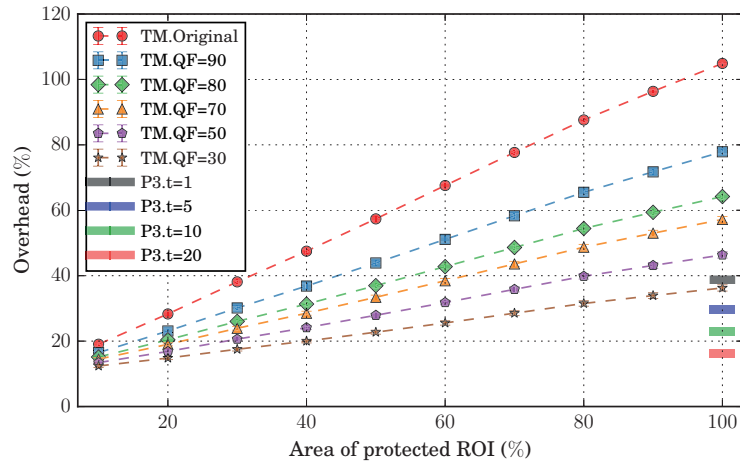
In addition, we also investigate the influence of quality factor in JPEG compression on storage overhead of JPEG Transmorphing. In this experiment, we used the USC SIPI dataset and compressed each raw image in JPEG using different quality factors in range of 30 and 100. For each quality factor, we performed JPEG Transmorphing in each image corresponding to ROIs of different sizes. We did not apply any overhead control in this experiment. In Figure 5.10, the storage overhead for each JPEG quality factor and different sizes of ROI (10% to 50%) is shown. From the results, one observes that the overhead in general decreases as the JPEG quality increases. This indicates that the overhead introduced by JPEG Transmorphing depends on not only the size of protection ROI but also the quality factor of the original JPEG image. This is also the reason that the overhead differs between the three datasets, as images in different datasets might be compressed in different quality factors.

### 5.2.2 Reconstruction Quality

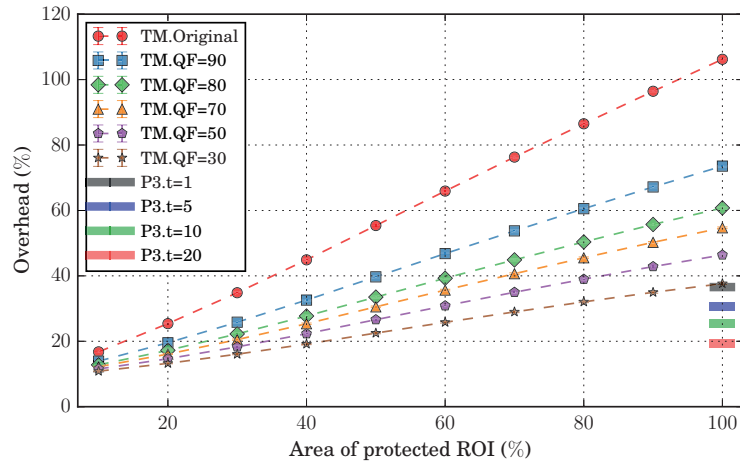
The second experiment aims to evaluate the quality of reconstructed images from Transmorphed images. For this experiment, we employed the subset of PIPA dataset [66] containing 1500 images<sup>2</sup>, defined in Section 4.2.1. In each image of the dataset, ground truth head annotations (rectangle) of identities are provided. In addition to the ground

---

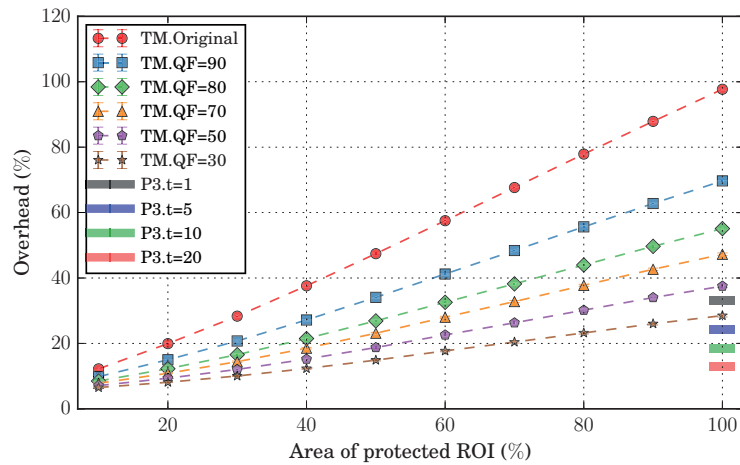
<sup>2</sup><http://grebvm2.epfl.ch/lin/thesis/dataset/PIPA-subset-1500.zip>



(a) USC SIPI dataset

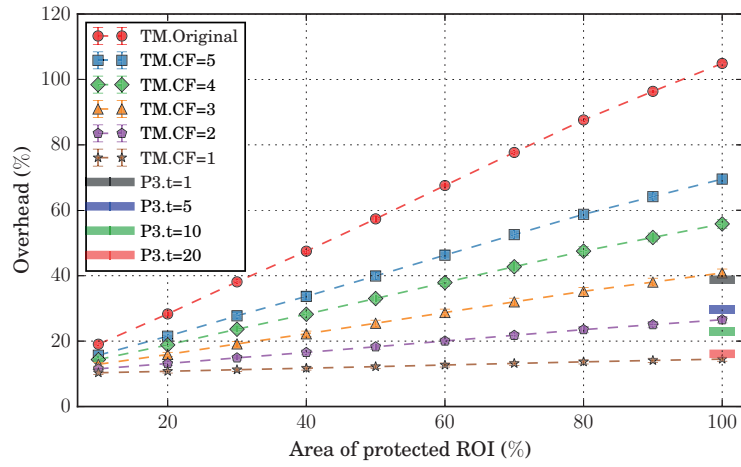


(b) PIPA dataset

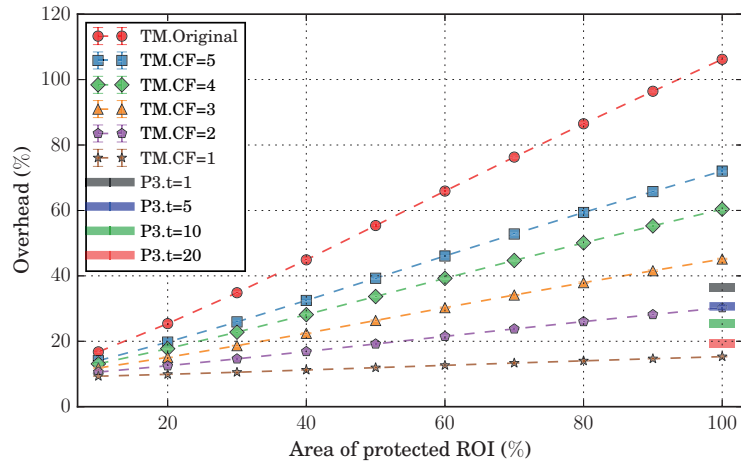


(c) INRIA Holidays dataset

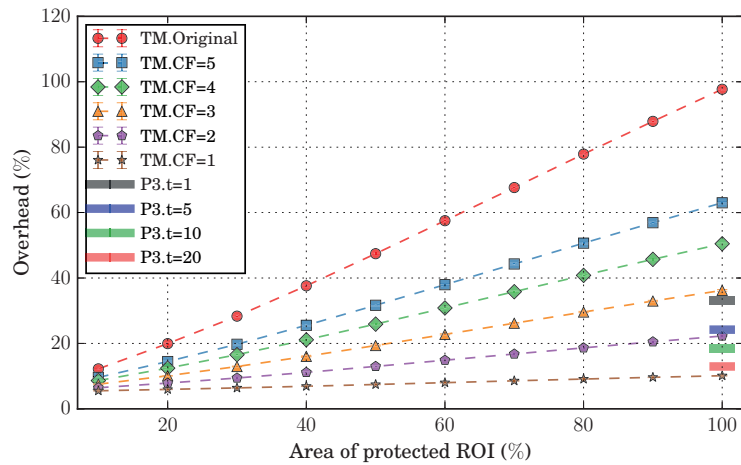
Figure 5.8 – Storage overhead of JPEG Transmorphing with overhead controlled by DCT Re-quantization.



(a) USC SIPI dataset



(b) PIPA dataset



(c) INRIA Holidays dataset

Figure 5.9 – Storage overhead of JPEG Transmorphism with overhead controlled by DCT Cut-off.

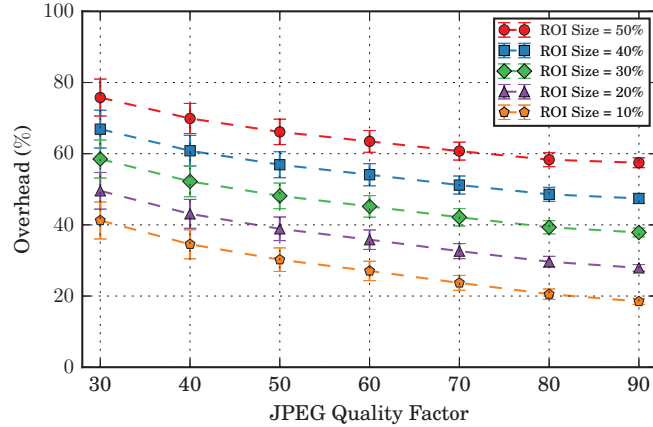


Figure 5.10 – Storage overhead of JPEG Transmorphing for images compressed with different JPEG quality factors.



Figure 5.11 – Illustration of three different image ROIs considered as protection targets.

truth **Head** region, we considered two more regions as protection target: (i) the **Full-body** region,  $3 \times$  head width and  $6 \times$  head height, with head at the top center of the full body and (ii) the **Upper-body** region, upper-half of the full body rectangle. The definitions of the three ROIs are illustrated in Fig. 5.11. Then, for each image, we applied JPEG Transmorphing on an identity’s three ROIs respectively, and applied P3 on the entire image<sup>3</sup> with a threshold of 5. For JPEG Transmorphing, we applied visual masking with a smiley sticker. Since the spatial-domain visual masking involves JPEG decoding and re-encoding, which affects the quality of reconstructed images, we encoded the masked JPEG images with the maximal JPEG quality factor of 100 to minimize such impact.

Firstly, executed the reconstructions from each protected image directly without transformation applied. Then, we applied four image transformations on each protected image by JPEG Transmorphing and P3 and reconstructed the images from the transformed images afterwards. The four transformations defined as follows:

<sup>3</sup>Original P3 algorithm only allows full image protection.

## Chapter 5. Secure JPEG Transmorphing

Table 5.1 – Mean PSNR (dB) and SSIM scores of images reconstructed from JPEG Transmorphing and P3, without and with image transformations applied.

Method	Without transformation	Reconstruction with transformation				
		Scaling	Cropping	Compression	Rotation	
Secure JPEG Transmorphing	PSNR:	45.08	39.37	42.94	37.59	45.08
	SSIM:	0.9874	0.9746	0.9790	0.0227	0.9874
P3	PSNR:	Inf.	37.46	43.27	35.53	Inf.
	SSIM:	1	0.9683	0.993	0.9253	1
Baseline JPEG compression (Q=75)			PSNR:		36.32	
			SSIM:		0.9444	

- **Scaling:** Subsample the image by a factor of 2 on both directions.
- **Cropping:** Crop the center region of the image with size of 512×384.
- **Compression:** Recompress the image in JPEG with a quality factor of 70.
- **Rotation:** Rotate the image by 90° in clockwise direction with a JPEG transcoder.

Finally we used two metrics to examine the quality of reconstructed image, compared to the original image: (i) Peak Signal-to-Noise Ratio (PSNR) and (ii) structural similarity index (SSIM) [73]. For scaling and cropping, the PSNR and SSIM were computed based on comparing the reconstructed image with the original image manipulated by the same scaling or cropping operation.

The mean PSNR and SSIM scores for different reconstruction conditions are shown in Table 5.1. For comparison, we also include the PSNR and SSIM of original images recompressed in JPEG with a quality factor of 75 as baseline. From the results, one observes that the average PSNR and SSIM of reconstructed images from JPEG Transmorphing are 45.08 dB and 0.9874, which would be considered practically lossless in the signal processing community. With different transformations applied on the Transmorphed images, the reconstructed images still preserve a high quality, with the PSNR and SSIM scores higher than that of the original images recompressed in JPEG with Q factor of 75. As a rotation operation by JPEG transcoding is lossless, the quality of corresponding reconstructed images is identical to that of the images reconstructed without transformation. For P3, the image reconstructed without transformation is lossless, which is because P3 algorithm directly performs on an existing JPEG image without JPEG decoding and re-encoding involved. With scaling and compression applied, the reconstruction quality of P3 is slightly worse than JPEG Transmorphing, but still could be still considered as lossless.

### 5.2.3 Privacy Protection Capability

Secure JPEG Transmorphing is proposed as an approach to preserve regional visual privacy in image. For many practical cases, such an approach is able to protect privacy perfectly, while still preserving the reversibility and high usability in the protected image. For instance, when a license plate or face region is completely hidden by an image sticker, there is no way to recognize them without any context information provided. Therefore, it is trivial to conduct simple evaluation such as the face recognition experiments conducted in Section 4.2. In a previous work by Oh et al. [44], the ability of several visual obfuscations for protecting image privacy is evaluated against automatic person recognition based on deep learning. Experimental results of the study reveal the fact that conventional regional privacy obfuscations like visual masking cannot ensure a perfect protection against automatic person recognition if similar context information is available in other public unprotected images of an identity. However, the degree to which different types of visual obfuscations can preserve users privacy against “attacking” from real human, especially in the context of online social media, has not yet been well studied. Therefore, we conducted a novel subjective experiment, where “attackers” were put in a simplistic social networking scenario, to evaluate the performance of different regional image obfuscations against person recognition from real human “attackers”.

#### User Study based on Crowdsourcing

The subjective experiment<sup>4</sup> was carried out by employing online subjects via Amazon Mechanical Turk (AMT)<sup>4</sup>, to recognize person in image obfuscated by different protection methods. To do so, we selected 6 identities (adult male) from the PIPA dataset [66] as the protection and recognition targets, each of which has four images, named *evaluation set*. We then applied seven types of obfuscations on each of the three ROIs (defined in Section 5.2.2) of the target identity in each evaluation image. The seven obfuscations are described in Table 5.2. In addition to the 6 identities, we selected another 3 identities to confuse subjects. Therefore, the subjects were required to recognize the protected identity in image from a total of 9 candidates. The lower the recognition rate, the stronger the method is able to preserve privacy. We assigned each identity an artificial name such that subjects can remember them easier.

**Evaluation Scenarios** Based on the image data, we designed three experiment setups, to simulate different scenarios of person recognition “attacks” in the context of online photo sharing:

- **Within-Context Person Recognition** In this scenario, we assume that an “attacker” has rich prior knowledge about the protected person in an image. The prior

<sup>4</sup><https://www.mturk.com/>

Table 5.2 – Visual privacy protection methods put in comparison in privacy evaluation.

Name	Description
SCRB.H	High-level JPEG Scrambling
SCRB.M	Medium-level JPEG Scrambling
P3.t=5	Regional P3 [32] protection with a threshold of 5
P3.t=20	Regional P3 [32] protection with a threshold of 20
Blur	Image blurring with radius of 8
Pixelate	Image pixelation with block size of 8
Mask	Visual masking using a smiley sticker (on head region) or a gray rectangle (on upper-/full-body region)

knowledge may be some public photos of the protected person in his/her online profile or from the attacker’s memory about the person if the attacker knows the person well or meets the person often. To model this scenario, we designed an experiment where subjects were provided with four reference images of each identity. The reference images are in original form without any visual obfuscation applied. We call them *reference set*. In this scenario, the identities in reference images have the similar or same context (dressing, event, people nearby or environment) as in their corresponding images of the evaluation set.

- **Across-Context Person Recognition** In the second scenario, we assume an “attacker” has also some prior knowledge about the protected person but the prior knowledge is less straightforward compared to the Within-Context scenario. For instance, the prior knowledge may come from other public photos of the protected person with different context compared to the protected images, e.g. a completely different photo album. For this scenario, we designed the second setup using a new set of reference images, which are very different from the evaluation set in the context of the target identity (e.g. dressing, event, people nearby or environment).
- **Without-Context Person Recognition** We also considered a scenario where the “attacker” has no direct prior knowledge about the protected person. However, the “attacker” may slightly know the person or has met the person once and therefore has a vague impression about the facial appearance of the protected person. To model this scenario, we designed the third setup where no any reference image is provided and subjects need to identify the protected person merely based on a profile picture showing the head of each identity.

The evaluation images and the two sets of reference images (Within-Context and Across-Context) of an example identity “Carl” are shown in Figure 5.12. The complete set of 24 evaluation images of six identities are shown in Figure A.3 in Appendix A. The reference images of all the 9 candidates for the two recognition scenarios (Within-Context and Across-Context) are shown in <http://grebvm2.epfl.ch/lin/privacy/ref/withinctx/images.html> and <http://grebvm2.epfl.ch/lin/privacy/ref/acxctx/images.html> respectively.





Figure 5.12 – Reference and evaluation image sets of an example identity.

We conducted subjective experiments based on AMT with the above three setups respectively. Each experiment was divided into three sessions, each showing a set of evaluation images with only one type of ROI protected. Thus we prevented any subject seeing the same image protected in different ROIs. Every HIT on AMT presents 6 protected evaluation images and an extra image unprotected serving as a “honeypot” to help us remove sloppy subjects. Since we have 168 different images ( $4 \text{ images} \times 6 \text{ identities} \times 7 \text{ obfuscations}$ ) for each protection ROI to evaluate, each session contains 28 different HITs (6 images/HIT). For Within-Context and Across-Context setups, the “honeypot” images are randomly selected from the reference images. While for Without-Context setup, “honeypot” images are only selected from the three extra identities apart from the six identities under evaluation. In such a way, the “honeypot” images do not reveal any information about original images the evaluation set. For Within-Context and Across-Context setups, the reference images of all 9 candidates are presented in the beginning of each HIT and were made always available during the experiment for subjects to review. Each HIT contains two questions, one for subjects to identify the protected person by selecting from the 9 candidates, the other acquiring the confidence about the subject’s answer. In the first question, if subject has no any clue about the protected person, he/she could choose “I really don’t know”, in which case the second question about recognition confidence does not appear. In the second question, we use 5-scale confidence scores, from 1 to 5, to indicate *Unsure*, *Not so sure*, *Neutral*, *Sure* and *Very sure* respectively. Figure 5.13 shows the screenshot of an evaluation image with the corresponding two questions presented in an HIT on AMT. The HITs of all the

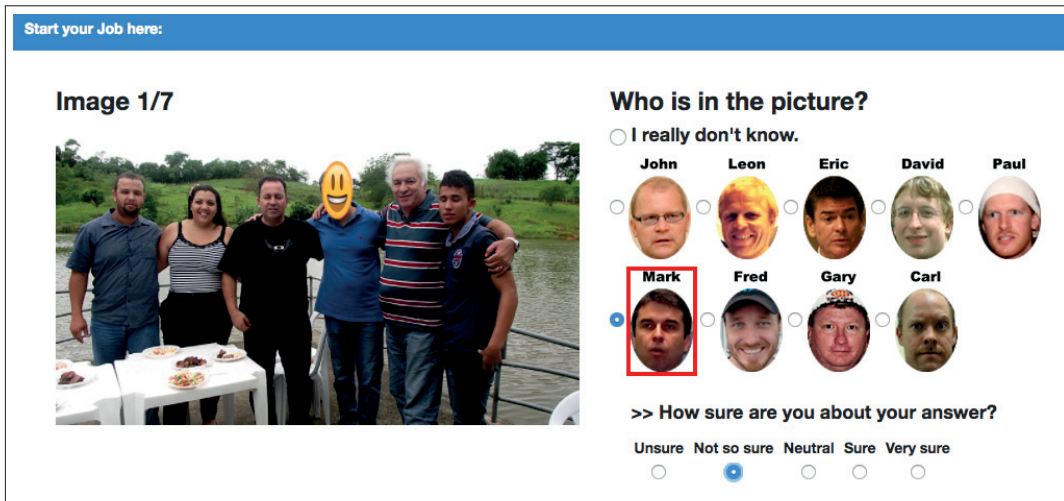


Figure 5.13 – Screenshot of an HIT presenting an image under subjective privacy evaluation on AMT.

experiment sessions were generated on AMT with the following constraints satisfied:

- The 6 evaluation images in each HIT belong to 6 different identities respectively;
- The order in which the 6 identities and applied obfuscations appear is random;
- Each image or HIT is rated by at least 7 different subjects;
- Each subject can take unlimited number of HITs within a session, but cannot participate in more than one session.

We filtered out outliers who answered the “honeypot” question incorrectly and those who always provided constant answers throughout all the questions and HITs. Finally, we collected answers from 241 subjects, each rating 39.2 images (including reference images) on average.

## Results and Analysis

Figure 5.14 shows the average proportion of correct, incorrect, and “I don’t know” answers over all images in the evaluation set, respect to different recognition scenarios and protection ROIs. First of all, one observes that the recognition accuracy (proportion of correct answers) of Within-Context scenario remains high in most cases: For images with only Head or Upper-body protected, the recognition accuracies against most obfuscations are above 60%, significantly higher than that of random guess, which is about 11.11% (1/9); While for Full-body protected images, the recognition rates are significantly reduced, but still well above the level of random guess. The performances of all the seven obfuscations

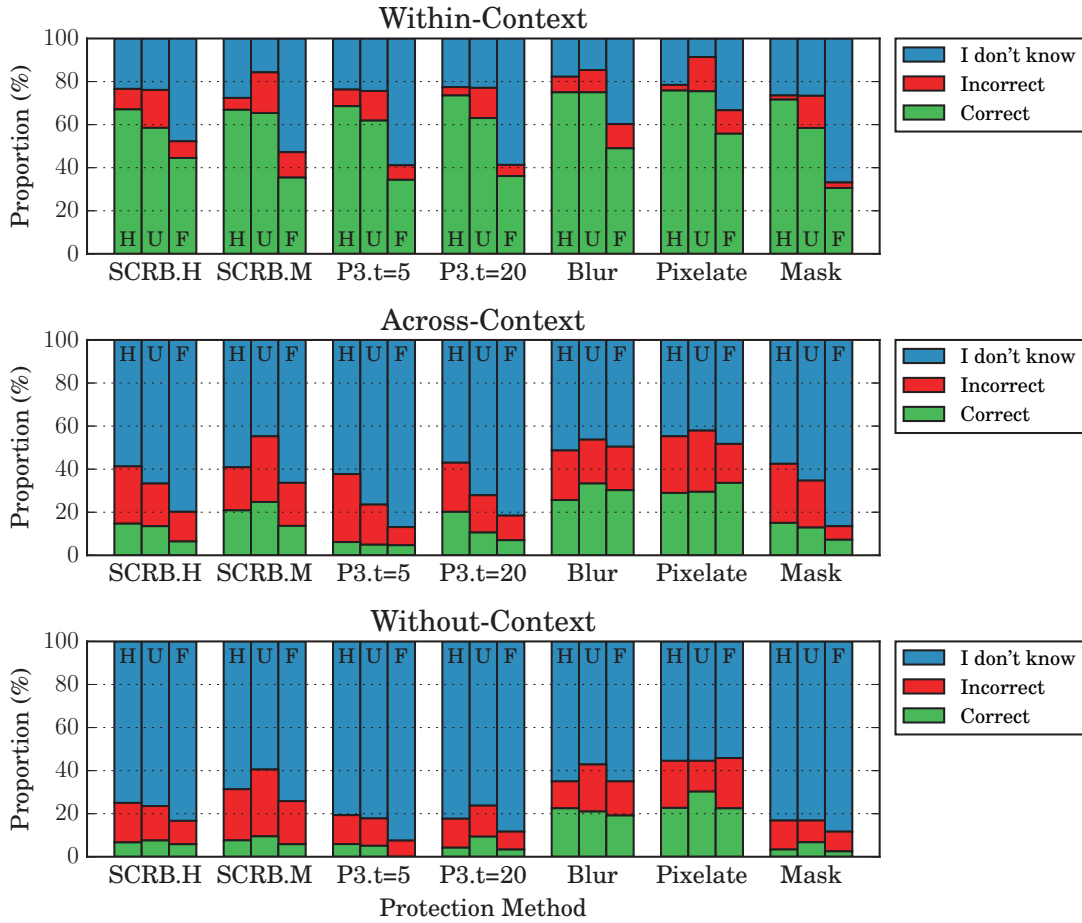


Figure 5.14 – Proportion of “I don’t know”, incorrect and correct answers across all images, with respect to different protection methods and regions. “H”, “U” and “F” annotated on each bar indicates Head, Upper-body and Full-body respectively.

are comparable, with Blur and Pixelate showing slightly worse protection to privacy than the others. We have to admit that in the case where direct context information about the protected person is available, most regional visual obfuscation cannot ensure a good level of privacy protection. In the scenario of Across-Context recognition, the overall recognition accuracies are greatly reduced. In turn, the proportions of incorrect and “I don’t know” answers are significantly higher. The recognition accuracies for most images protected by methods such as JPEG Scrambling, P3 and visual masking are lower than 20%, close to the level of random guess accuracy. This time, one observes that recognition accuracies against Blur and Pixelate protections are obviously higher than the other methods, which is because the two methods still disclose certain amount of low-resolution visual information about the original image region. In images protected by Blur and Pixelate, size of protection ROI does not show a significant impact on the accuracy of recognition. As for the Without-Context scenario, recognition accuracies

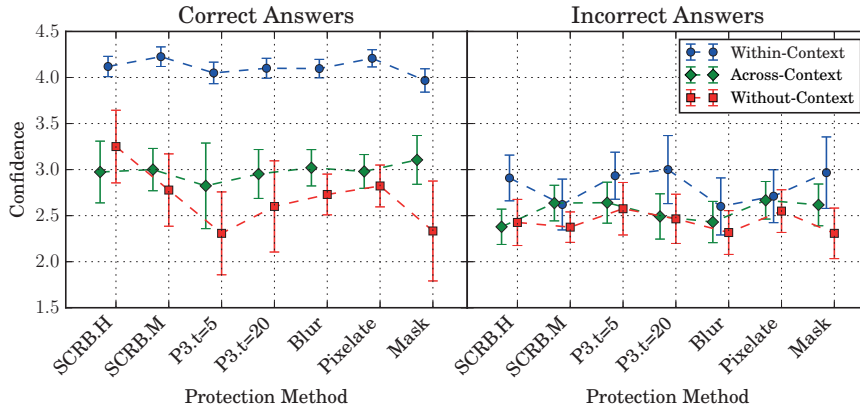


Figure 5.15 – Certainty scores of correct and incorrect recognition answers with respect to different scenarios and protection methods.

are further reduced: the proportions of correct answers for protection methods including JPEG Scrambling, P3 and visual masking are all below 10%. However, the accuracies for Blur and Pixelate are still higher than 20%. Among all the seven obfuscations, Mask and P3.t=5 provide the strongest protections, which is reasonable as masking operation completely hides the visual information behind the mask and a strong level of P3 protection is visually extremely to a gray mask.

We then investigate how different visual obfuscations affect the subjects recognition confidence in different conditions. We consider subjects confidence in two dimensions: the confidence of correct recognition answers and the confidence of incorrect answers. For both correct and incorrect answers, we plot in Figure 5.15 the confidence scores (mean and 95% confidence interval) corresponding to each recognition scenario and protection method. For Within-Context scenario, the confidence scores of correct answers are mostly above 4.0, significantly higher than the other two scenarios. The overall confidence of correct recognition in Across-Context scenario is around 3.0, a *neutral* level between *sure* and *not so sure*. In the Without-Context scenario, the confidence scores of correct answers are slightly lower than that of Across-Context, falling down to the level of *not so sure*. In addition, the confidence scores vary between applied obfuscations: For correct answers in Within-Context scenario, the overall confidence of Mask protection is the lowest; High-level JPEG Scrambling (SCR.B.H) and P3 with a small threshold of 5 (P3.t=5) also generate slightly lower confidence scores than the remaining methods. As for incorrect recognition answers, the confidence scores are lower than 3.0 in most cases. This is natural as subjects tend to generate wrong decisions if they are not confident enough. However, the confidence scores of incorrect answers corresponding to different obfuscation methods and recognition scenarios show relatively random patterns, where the confidence in Within-Context scenario is just slightly higher. From the results, some obfuscations (e.g. SCR.B.H, P3 and Mask) result in relatively higher confidence for incorrect recognition answers, implying that those methods could perform better in “confusing” subjects.

## 5.2. Performance Evaluation

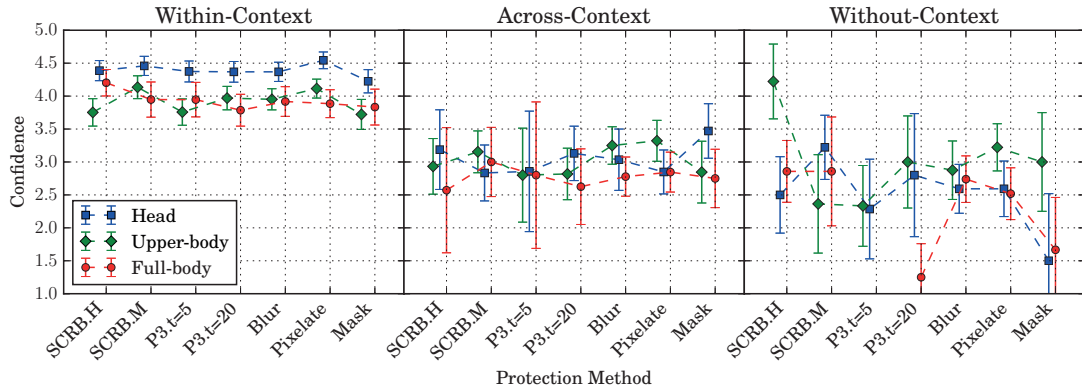


Figure 5.16 – Certainty scores of correct recognition answers with respect to different scenarios, protection ROIs and protection methods.

At the end, for each scenario, we also plot the confidence scores of correct answers corresponding to each protection ROI and protection method, shown in Figure 5.16. In the scenario of Within-Context recognition, the confidence scores are the highest in Head region protected images, for all the seven obfuscations. While, the confidence scores for Upper-body and Full-body protected images are still higher than 3.5, and comparable. For Across-Context and Without-Context scenarios, the overall recognition confidence is much lower, consistent with the results observed in 5.15. However, one cannot clearly distinguish the difference in confidence between different protection ROIs for the two scenarios, which is also consistent with our findings in Figure 5.14, where the recognition accuracies for the three ROIs are not significantly different. Due to very few correctly recognized samples in the two scenarios, the confidence interval of confidence scores is large.

The experimental results indicate that the contextual information such as unprotected parts of human body, clothes and people nearby the protected person indeed can provide some visual cues for real humans to recognize protected person in an image. This is why the recognition accuracy for the Within-Context scenario is much higher than the other two scenarios, in which the “I don’t know” and incorrect answers predominate. This means certain regional obfuscations can well conceal the identity of a protected person if not much meaningful context information is available. By comparing the recognition results obtained on three different protection ROIs, Full-body obfuscations always provide the strongest protection to privacy, as it reveals the minimum amount of information about the human body. The recognition accuracies against Head and Upper-body obfuscations are comparable and both higher than that of Full-body protection. This implies that even if disclosing some parts of the person’s body, “attackers” may utilize such information to recognize the obfuscated person.



Figure 5.17 – 13 images used in pleasantness evaluation of privacy protection methods.

### 5.2.4 Pleasantness

As the last experiment, we attempt to understand the pleasantness with respect to users’ perception and usage preference of different visual privacy protection methods. Hence, in this experiment, we define the term “pleasantness” in twofold:

- The **Perception Pleasantness**, which refers to users emotional perception when observing others photos protected by a particular method.
- The **Usage Pleasantness**, which refers to users preference to use a particular method to protect their own privacy in online photo sharing.

We also employed the Amazon Mechanical Turk (AMT)<sup>5</sup> to conduct subjective experiments based on crowdsourcing. We selected 13 images from the PIPA dataset [66] (shown in Figure 5.17), and applied 10 visual protections on a person’s head region in each image. Therefore, it resulted in 130 different visually obfuscated images<sup>6</sup>. The applied visual protection methods are listed in Table 5.3. An example image obfuscated by the 10 visual protections is illustrated in Figure 5.18. To measure perception pleasantness, we apply the Valence model in psychology using 9-Point SAM Scales [76], where 1 stands for very unpleasant, 9 for very pleasant and the middle point 5 for neutral. As for usage pleasantness, we use the three-level preference scales, i.e. “Dislike”, “Neutral” and “Like”. We removed the results from one subject who is considered as outlier as he/she provided very constant answers. The screenshot of an HIT on AMT for this experiment is given in Figure A.4 in Appendix A. On AMT, we asked 25 different subjects to vote on the pleasantness of each of the 130 protected images. Finally, 105 unique subjects participated in our experiment, each voting on 30.95 images on average.

---

<sup>5</sup><https://www.mturk.com/>

<sup>6</sup>All the 13 images and their protected versions used in the experiment are available at [http://grebvm2.epfl.ch/lin/privacy/dataset\\_privacy\\_pleasantness.zip](http://grebvm2.epfl.ch/lin/privacy/dataset_privacy_pleasantness.zip).

Table 5.3 – Visual privacy protection methods being compared in pleasantness evaluation.

Name	Description
SCRB	High-level JPEG Scrambling
P3	Regional P3 [32] protection with a threshold of 20
Pixelate	Image pixelation with block size of 20
Blur	Image blurring with radius of 20
Black	Visual masking in black color
Smiley	Visual masking with a “Smiley” Emoji <sup>7</sup>
TearsJoy	Visual masking with a “Face with Tears of Joy” Emoji, the 2015 word of the year by Oxford English Dictionary <sup>8</sup>
SnapGhost	Visual masking with a Snapchat Ghost logo <sup>9</sup>
Vendetta	Visual masking with a cartoon Guy Fawkes mask originally from the film <i>V for Vendetta</i> <sup>10</sup>
C-Stamp	Visual masking with a gray stamp showing “CONFIDENTIAL”

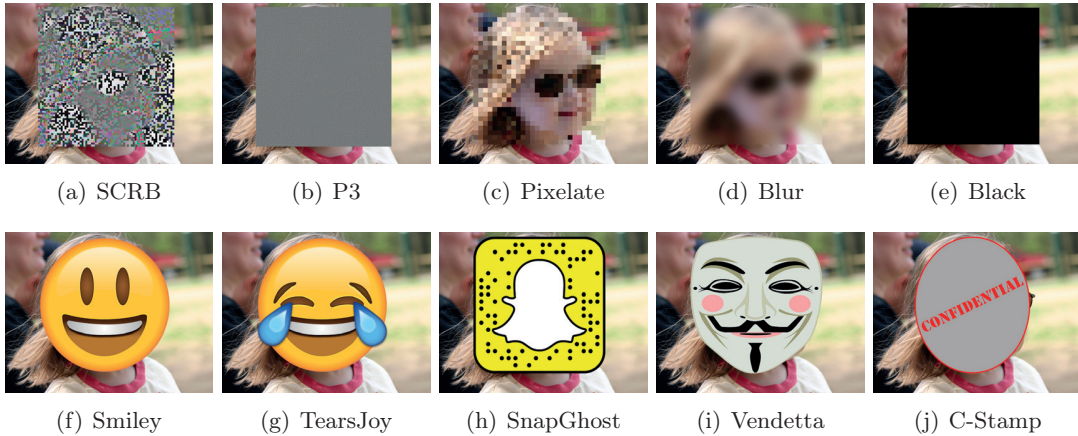


Figure 5.18 – 10 different visual privacy protection methods.

## Results and Analysis

**Perception Pleasantness** The overall results on perception pleasantness of the ten protection methods are given in Figure 5.19, which shows the average perception pleasantness score and its 95% confidence interval corresponding to each protection method. From the result, one observes that different methods reveal significantly different degrees of perception pleasantness. Among the ten methods, Pixelate, Blur, Smiley and TearsJoy Emoji provide obviously higher pleasantness scores than the others: The mean pleasantness scores of Pixelate, Blur, Smiley and TearsJoy are all above 5.0, indicating positive emotions; While others reveal only negative pleasantness with average pleasantness scores

<sup>7</sup>Emoji sticker downloaded from <https://emojiland.com/pages/free-download-emoji-icons-png>

<sup>8</sup><http://blog.oxforddictionaries.com/2015/11/word-of-the-year-2015-emoji/>

<sup>9</sup><https://www.snap.com/en-US/brand-guidelines/>

<sup>10</sup>[https://en.wikipedia.org/wiki/V\\_for\\_Vendetta\\_\(film\)](https://en.wikipedia.org/wiki/V_for_Vendetta_(film))

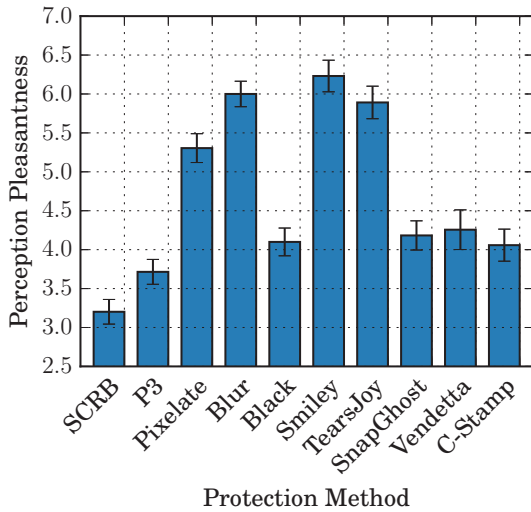


Figure 5.19 – Overall perception pleasantness scores of different protection methods.

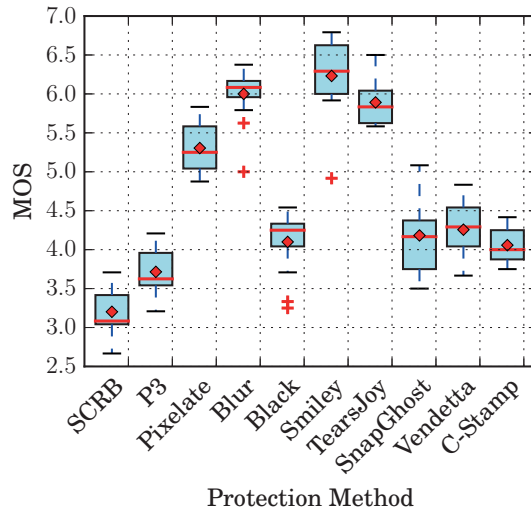


Figure 5.20 – Distribution of MOS across all 13 images for each protection method.

below 5.0. Among the ten methods, Secure JPEG Scrambling (SCRIB) and P3 provide the lowest pleasantness, lower than 3.8. We believe this is because image pixelation and blurring generate the most natural visual effects compared to the others while the two Emoji stickers are the most enjoyable and amusing ones among all the methods. On the contrary, the two distortion-based approaches, JPEG Scrambling and P3, result in the most ugly visual effects. Interestingly, image stickers such as the SnapGhost and Vendetta, though funny and interesting, still reveal relatively low pleasantness. The other two methods, Black masking and C-Stamp, generate similar levels of pleasantness as SnapGhost and Vendetta. Then, for each of the 130 images, we computed its mean pleasantness score (MOS) across the ratings of different subjects. For each protection method, we plot the mean pleasantness scores of all the 13 images in a box and whisker diagram in Figure 5.20. In this figure, each box plot represents the spread of mean pleasantness scores over different image content. The results basically coincide with the observations in Figure 5.19, revealing a significant difference in pleasantness between the 10 protection methods. In addition, the box plots also reveal somewhat influence of image content on pleasantness, although such an impact is not significant: For most of the protection methods, the maximum difference in pleasantness scores between images is in range of 1 and 1.5; Also, for a certain method, the pleasantness scores of different images do not usually change drastically across the boundary (5.0) between pleasant and unpleasant emotions.

**Usage Pleasantness** We then analyze the results on usage pleasantness measured by the three preference votes. For each protection method, we first compute the overall proportions of votes for “Dislike”, “Neutral” and “Like” respectively. The results are shown



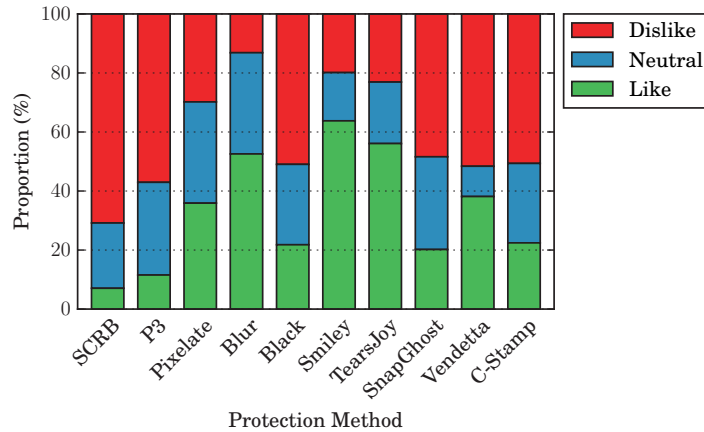


Figure 5.21 – Histograms of “Dislike”, “Neutral” and “Like” for different protection methods.

in Figure 5.21. Again, the two Emoji stickers Smiley and TearsJoy obtain the most votes for “Like”, above 60% and 55% respectively. For the two methods, about 20% subjects voted for “Dislike” and 20% for “Neutral”. Similar as the observations in perception pleasantness, Pixelate and Blur also received a large number of votes for “Like” (35% and 51%), and in the same time a large number of votes for “Neutral” (both around 30%). Particularly, the Blur protection received the minimum proportion of votes for “Dislike”, which might due to its high visual naturalness. This time, the Vendetta mask also got a considerable proportion of votes for “Like” with the smallest proportion of “Neutral” among all the 10 methods. Compared to SnapGhost and C-Stamp, Vendetta got the same number of votes for “Dislike”. This reveals that the Vendetta mask is prone to being either liked or disliked by people. In addition, the other methods all received much less votes for “Like”. Among all the 10 methods, the two distortion-based obfuscations, JPEG Scrambling and P3 are the least preferred by people for usage.

**Perception pleasantness vs. Usage pleasantness** Comparing the results between perception pleasantness and usage pleasantness, one observes a significant correlation. To investigate how the two types of pleasantness correlate with each other, we make the scatter plots in Figure 5.22, showing the the perception pleasantness MOS versus the proportions of “Dislike”, “Neutral” and “Like” votes of each image respectively. In addition, two correlation metrics are computed to quantify the degree of correlation: the Pearson correlation coefficient<sup>11</sup> [77] and the Spearman’s rank correlation coefficient<sup>12</sup>. As Figure 5.22 shows, the perception pleasantness (MOS) is highly correlated with the two votes for “Dislike” and “Like”, but not obvious for “Neutral”. The two correlation scores between perception pleasantness and proportion of “Like” are both higher than 0.9. The absolute values of two correlation scores between perception pleasantness and “Dislike”

<sup>11</sup><https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.pearsonr.html>

<sup>12</sup><https://docs.scipy.org/doc/scipy-0.15.1/reference/generated/scipy.stats.spearmanr.html>

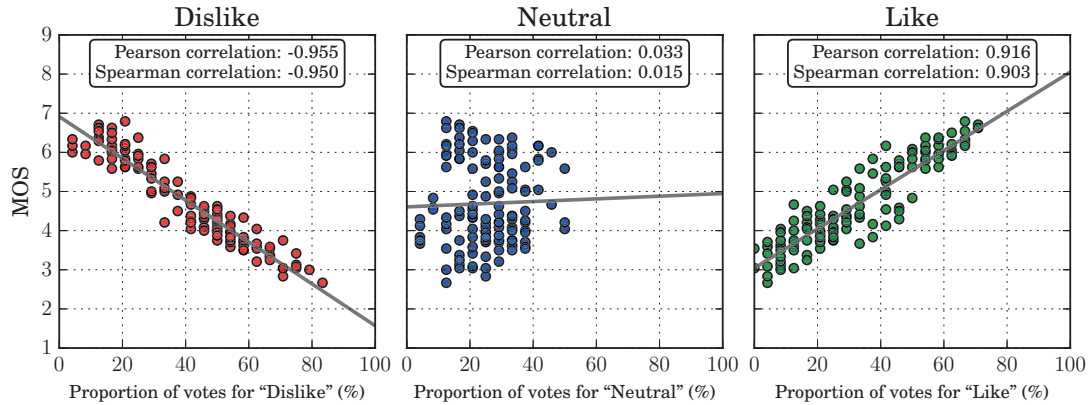


Figure 5.22 – Correlation between perception pleasantness (MOS) and proportion of votes for three different preference options.

are both greater than 0.95, but representing a strong negative correlation. However, the proportion of votes for “Neutral” is not obviously correlated to perception pleasantness, where both correlation scores are close to zero.

### 5.3 Discussions

**Security** In the proposed JPEG Transmorphing algorithm, sensitive privacy information is considered to be associated in the sub-image, which is encrypted before being inserted in the protected JPEG image. Therefore, the security of the proposed algorithm mostly relies on the encryption algorithm applied on the sub-image. In practice, the state of the art algorithm can be used to encrypt the sub-image, e.g. Advanced Encryption Standard (AES) [78]. The security analysis is out of the scope of this thesis.

**Privacy** The primary purpose of using JPEG Transmorphing is to protect regional visual privacy in image, in a reversible and personalized manner. Frankly, obfuscation of partial image regions may not perfectly preserve privacy revealed from other unprotected regions, as is shown in Section 5.2.3 and [44]. Yet, we still consider it as a powerful tool to protect privacy in cases where meaningful context information is unavailable. As our experiments in Section 5.2.3 imply, if “attackers” do not possess considerable amount of relevant context information about the protected person, hiding just a head region can significantly decrease the accuracy and confidence of person recognition. We also admit that our experiments have certain limitations to perfectly simulate the realistic scenarios. For example, in the experiments, subjects only needed to identify the person from 9 candidate identities, which already provides a 10% probability to get the correct answer by just a random guess. Besides, we had to explicitly tag each identity in reference images so that subjects could conduct the survey. It is however not easy in reality for

Table 5.4 – Qualitative comparison of different reversible visual privacy protection methods. Sc., Cr., Co. and Ro. denote four types of image transformations, namely scaling, cropping, JPEG compression and rotation respectively.

Method	Modified Domain	Level of Overhead	Visual Effect	Regional Protection	Compatibility with Image Transformation			
					Sc.	Cr.	Co.	Ro.
Poller [31]	Pixel	Median (<160%)	Distorted	✓	✓	×	✓	×
P3 [32]	DCT	Low (<40%)	Distorted	✓	✓	✓	✓	✓
Cryptagram [33]	Bitstream	Varied (40 - 680%)	Personalized	×	×	×	×	×
PUPPIES [19]	DCT	Varied (50 - 800%)	Distorted	✓	✓	✓	✓	✓
Wright [15]	Pixel	Varied (100 - 800%)	Distorted	×	×	×	✓	✓
Sun [18]	DCT	Low (<6%)	Distorted	×	✓	×	✓	✓
JPEG Scrambling	DCT	Low (<20%)	Distorted	✓	×	×	×	✓
JPEG Transmorphing	DCT	Median (<120%)	Personalized	✓	✓	✓	✓	✓

an “attacker” to know which set of public images belong to the protected person, among all the public images he/she can access. Therefore, in reality, recognition of a protected person from merely an image without direct context information available might be more difficult.

**Usability** In fact, the JPEG Transmorphing employs the similar idea as P3, where two portions of the same image are generated: a public portion that can be available to any party and a secret portion considered as secret and privacy-sensitive. The most significant difference is that, JPEG Transmorphing preserves the secret portion, namely the sub-image, in the protected file itself (the public portion) in form of APPn markers. While P3 splits the two portions in two separate files and employs another server to manage the secret image. Compared to JPEG Transmorphing, the drawbacks of P3 are twofold: (i) It creates two separate files, which potentially complicates the file management system. (ii) P3 generates only the grayish visual effects in its public images, which basically has no difference from any other distortion-based approach in the pleasantness point of view. This brings us to the discussion on the usability of our proposed method. With JPEG Transmorphing, one can protect arbitrary image regions using most types of spatial-domain manipulations, including not only conventional filters like blurring, pixelation and warping, but also various interesting image manipulations such as inpainting, sticker addition, and image style transfer [79]. It provides a significant flexibility and usability such that users can choose their preferred ways to protect any sensitive image regions while preserving the reversibility of the protected image. This is the most distinctive

characteristic of our method compared to the others. Although such a high usability is at the cost of file size expansion in the Transmorphed image, the file size can be adjusted by the proposed approaches to overhead control or fine-grained definition of protection ROIs.

To summarize our discussions, a qualitative comparison of selected reversible methods for visual privacy protection is given in Table 5.4. From this comparison, one can observe the advantage of the proposed JPEG Transmorphing over other methods: Using JPEG Transmorphing, reversibility of protected images can be achieved at the cost of a moderate level of storage overhead. At the same time, image protected with JPEG Transmorphing is robust to most image manipulations. While, most all methods cannot fulfill all the above features.

### 5.4 Conclusion

This chapter presents secure JPEG Transmorphing, a flexible framework for protecting image visual privacy in a secure, reversible and personalized manner. The working principle of JPEG Transmorphing is using JPEG application segments markers (APPn) to secretly preserve partial original image information, while encoding the “cover” JPEG image in a visually protected form. The original image visual information can be protected by almost any image obfuscation, such as visual masking, blurring, pixelation, inpainting and warping. The protected image (Transmorphed image) has the same syntax as standard JPEG, and is therefore backward compatible with JPEG. With a dedicated JPEG transcoder or decoder that supports JPEG Transmorphing, the original image can be recovered by replacing the obfuscated regions in the protected image with the corresponding original regions extracted from APPn markers. Performances of the proposed method have been evaluated and studied with a set of experiments. Experimental results show that the images protected with JPEG Transmorphing are robust to most lossy and lossless image transformations like scaling, rotation, cropping and compression. Although a Transmorphing protection causes overhead to image file size, such overhead can be modulated by the proposed overhead control mechanisms, without affecting the reconstruction quality. In addition, with two subjective experiments conducted via online crowdsourcing, the proposed method shows great potentials to provide a good degree of privacy preservation, and much higher usability from a subjective pleasantness point of view than JPEG Scrambling and P3. We admit that regional image obfuscation may not offer the perfect protection to privacy as unprotected regions in image may still reveal private information. Yet, it is useful enough in many practical scenarios of online photo sharing. For instance, when one attempts to protect his/her children or a license plate in image from public but still hopes to share the original image with intimate connections, the proposed secure JPEG Transmorphing is well competent for the job.

# The Architectures **Part II**



## 6 ProShare: Privacy-Preserving Photo Sharing based on a PKI

OSN sites usually offer conditional access as a mechanism to help users protect their own privacy. Conditional access works based on the fact that users need to trust the service provider to enforce the access control based on user-defined policy. Irrespective of a service provider's sincerity in matters of privacy, all image sharing platforms exhibit the same basic flaw: Once an image has left the device it was created on, its owner loses control over who will have access to his image, when and where. Researchers have proposed different approaches to preserve privacy by secure protection of image content. Here, the protection and reconstruction are based on a secret key and most of the proposed approaches can be seen as a "symmetric encryption", where one of the central issues is how to exchange keys between subjects. However, in most proposed approaches, this issue is not addressed and most studies simply assume that users exchange symmetric keys in an offline secure channel.

In this chapter, we present ProShare, an architecture for a privacy preserving service applicable to image protected by Secure JPEG. The architecture is built based on a public key infrastructure (PKI) integrated with a ciphertext-policy attribute-based encryption (CP-ABE). In ProShare, a photo is securely protected by a Secure JPEG protection algorithm with a secret key. The secured photo is then safely kept on an untrusted service (server, cloud, etc.). Meanwhile, the secret key is encrypted by CP-ABE with a user-defined access policy. With the help of the PKI, users can share ABE private keys between each other and only those who possess the right ABE private keys associated with matched attributes are able to recover the original image.

The rest of the chapter is structured as follows. Section 6.1 outlines the fundamental cryptographic protocols flowing into this work. Section 6.2 describes in detail the system design with particular emphasis on the use of different cryptographic protocols and how these are integrated to provide a consistent and secure process flow. Section 6.3 presents our implementation of a demonstrator based on the proposed architecture. Finally, Section 6.4 concludes this chapter.

### 6.1 Cryptography Basics

This section provides the fundamentals of cryptography protocols that are used in the proposed privacy-preserving photo sharing architecture. It starts with the conventional public key cryptography and infrastructure followed by a special type of public key encryption in which the secret key of a user and the ciphertext are dependent upon attributes, namely attribute-based encryption.

#### 6.1.1 Public Key Cryptography and Infrastructure

Secure protection of an image powered by Secure JPEG can be considered a form of symmetric encryption, where the same secret key is used to protect (encrypt) and recover (decrypt) the image. Therefore, securely exchanging secret keys with different entities, more precisely, the people who are authorized to see the original image, is vital and challenging. Public key cryptography (PKC) provides a reliable and efficient way to exchange secrets securely. PKC is an asymmetric cryptography, where a pair of keys is used to encrypt (using public key) and decrypt (using private key) a message respectively. Any user who wants to share a secret message obtains the public key of the intended recipient, encrypts the message using this key and sends it to the recipient. On the other side, the recipient uses his private key to decrypt the encrypted message. Therefore, the public key can be available to public and does not reveal any secret information. While the private key should be kept securely by its owner so that no one else can decrypt a message that is encrypted by his/her public key. Various public key algorithms have been proposed, among which RSA [80] is the most widely used.

Authentication of a public key is the central problem of using public-key cryptography, i.e. how to prove a public key belongs to the right person or entity claimed, or has not been tampered with or replaced by a malicious third party. The usual solution to this issue is the public key infrastructure (PKI), an arrangement that binds public keys with respective identities of entities (people or organizations). The binding process is established through registration and issuance of digital certificates by a certificate authority (CA). A digital certificate certifies the ownership of a public key by the named subject of the certificate, which allows others to rely upon signatures or on assertions made about the private key that corresponds to the certified public key. The CA digitally signs and publishes the public key bound to a given user, using the CA's own private key. Therefore, the CA acts as a trusted third party (TTP): trusted both by the owner of the certificate and by the entities relying upon the certificate.

#### 6.1.2 Attribute-Based Encryption

Attribute-based encryption (ABE) is a relatively new cryptography approach that revises the concept of conventional public key cryptography. As the name implies, ABE brings



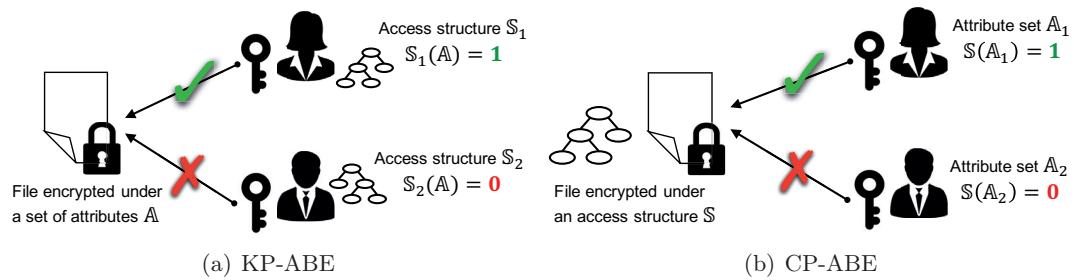


Figure 6.1 – Illustration of two schemes of attribute-based encryption.

the notions of *Attributes* and *Policy* during the encryption and decryption procedures. The principle of ABE is to enable a ciphertext to be decrypted by multiple secret keys possessed by different entities, as long as the secret keys and the ciphertext have a good match in terms of attributes and an access policy ABE defines an identity not atomically but as a set of attributes about the identity, e.g. the age, role and relationship. An attribute in ABE can be either a nominal (e.g. ‘Close Friend’ or ‘Name:Alice’), a numerical (e.g. ‘age  $\geq 18$ ’ or ‘ID = 6’) or a negation (e.g. ‘NOT Name:Bob’) expression. A policy is an access structure over the universe of attributes, constructed using conjunctions, disjunctions or  $(k, n)$ -threshold gates. A typical example of access policy is (‘Close Friend’ OR ‘Family’ OR (‘age  $> 16$ ’ AND ‘age  $\leq 25$ ’ AND (NOT ‘Bob’))). Depending on the association of attributes and access structure, two types of ABE schemes have been developed: the Key-Policy Attribute-Based Encryption (KP-ABE) [81] and the Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [82]. We use two diagrams in Figure 6.1 to illustrate the working principle of KP-ABE and CP-ABE respectively.

In **KP-ABE**, the access policy is encoded in each entity’s private key, and the ciphertext is associated with a set of attributes. A private key is able to decrypt a ciphertext only if the set of attributes in the ciphertext satisfy the policy integrated in the key. In KP-ABE, the access policy is enforced on the private keys of users and therefore the encryptor is considered to exert no control over who can access the data he/she encrypts [82], except by his choice of descriptive attributes for the data. The encryptor has to trust the key-issuer to issue appropriate keys to grant or deny access to the appropriate users. In this regard, the “intelligence” of KP-ABE is assumed to be with the key-issuer, rather than the encryptor.

In **CP-ABE**, the access policy is integrated in the ciphertext when encrypting a message, and a user’s private key is associated with a set of attributes. If the attributes in a private key satisfy the policy associated in a ciphertext, the private key is able to decrypt the ciphertext. Therefore, CP-ABE allows to realize an implicit access control, where authorization is included in encrypted data and only those who are given attributes-matched private keys can access the original data. Contrary to KP-ABE, the

“intelligence” of CP-ABE is exerted on the data-encryptor, rather than the key-issuer. Another advantage of CP-ABE over KP-ABE is that users can obtain their private keys after data has been encrypted. The data can be encrypted by only specifying the access policy that allows to decrypt it, without knowing the actual set of users who may have access. Any future user that will be issued a key with respect to attributes satisfying the policy will be able to decrypt the data. A CP-ABE protocol usually consists of the following fundamental algorithms or operations [82]:

1. **Setup:** The initial setup procedure takes as input implicit security parameters (e.g. random seed initialized by a user ID or name) and generates a pair of keys for a user: an ABE public key ( $APK$ ) and an ABE master secret key ( $AMSK$ ).
2. **Key Generation** ( $AMSK, \mathbb{A}$ ): This algorithm takes as input the ABE master secret key  $AMSK$  and an attribute set  $\mathbb{A}$  to generate an ABE private key  $ASK$ .
3. **Encryption** ( $APK, m, \mathbb{S}$ ): The encryption algorithm encrypts a message  $m$  to its ciphertext  $C_m$ . Input parameters include the ABE public key  $APK$ , the message  $m$  and an access structure defined by  $\mathbb{S}$ .
4. **Decryption** ( $APK, C_m, ASK$ ): The decryption algorithm takes as input the ciphertext  $C_m$ , the ABE public key  $APK$  of the encryptor, and an ABE private key  $ASK$  of the decryptor. Only if the set of attributes  $\mathbb{A}'$  associated in  $ASK$  satisfy the access structure  $\mathbb{S}$  implicitly defined in the ciphertext  $C_m$ , i.e.  $\mathbb{S}(\mathbb{A}') = \text{True}$ , will the original message  $m$  be successfully decrypted.

Due to the above stated reasons, we employed CP-ABE in the proposed ProShare photo sharing architecture. The security proofs including more detailed mathematics of CP-ABE algorithms is given in [82, 83] and therefore is not covered in the thesis.

## 6.2 ProShare: The Architecture Design

In this section we describe in detail the architecture design of ProShare. All notations with corresponding descriptions used in this chapter are listed in Table 6.1. To better understand the following, we first distinguish two types of roles in such a photo sharing system: (i) the *sender*, who wants to post and share a photo with friends; (ii) the *requester*, who attempts to review a photo shared by the sender.

### 6.2.1 Operating Principle

First of all, we define the following functional requirements for the proposed photo sharing architecture: A sender can select and then protect an arbitrary set of regions in a given photo, using any preferred visual obfuscation; Once protected, the resulting image file can

Table 6.1 – Notations used in describing ProShare architecture.

Notation	Description
$u_i$	A user identified by $i$
$u_i.SS.KS$	$u_i$ 's storage service (SS) on key server (KS)
$u_i.SS.CS$	$u_i$ 's storage service (SS) on content server (CS)
$(m, C_m)$	A pair of plaintext $m$ and its ciphertext $C_m$
$(I_O, I_P)$	A pair of original image $I_O$ and its protected form $I_P$
$(TPK, TSK)$	A pair of PKC public key $TPK$ and private key $TSK$
$(APK, AMSK)$	A pair of ABE public key $APK$ and master secret key $AMSK$
$ASK_{i \rightarrow j}$	ABE private key $ASK$ for user $j$ issued by user $i$
$JPEGSec(I_O, \mathbb{P})$	The function to protect an image $I_O$ using a Secure JPEG protection algorithm defined by a parameter set $\mathbb{P}$ , which specifies the algorithm type (Scrambling or Transmorphism), secret key and algorithm specific parameters, e.g. mask matrix, strength (Scrambling) or/and encryption scheme (Transmorphism)
$JPEGRec(I_P, K)$	The function to recover the original image from a secure image $I_P$ with a secret key $K$
$PKCSetup()$	The function to generate a pair of PKC public key $TPK$ and PKC private key $TSK$
$PKCEnc(m, TPK)$	The function to encrypt a message $m$ in PKC with a public key $TPK$
$PKCDec(c, TSK)$	The function to decrypt a ciphertext $c$ in PKC with a private key $TSK$
$ABESetup()$	The function to generate pair of ABE public key $APK$ and ABE master secret key $AMSK$
$ABEKeyGen(\mathbb{A}, AMSK)$	The function to generate an ABE private key $APK$ with an attribute set $\mathbb{A}$ and an ABE master secret key $AMSK$ as input
$ABEEnc(m, \mathbb{S}, APK)$	The function to encrypt a message $m$ in CP-ABE with an access structure $\mathbb{S}$ and an ABE public key $APK$ as input
$ABEDec(c, ASK, APK)$	The function to decrypt a ciphertext $c$ in CP-ABE with an ABE private key $ASK$ and an ABE public key $APK$ as input

be freely shared and viewed with any JPEG compliant decoder; The shared image file is internally consistent and therefore contains all the data necessary to view both protected and unprotected image regions; Finally, the photo sender can dynamically associate an access policy with the protected photo and assign a set of attributes with prospective requester. Only if the photo sender has granted a requester a set of matching attribute(s) compared to the access policy defined for the photo, will this photo be accessible to that particular requester. Complying with the above attributes, such a service allows for the protection of privacy in photos prior to them leaving the user's device. Once shared, the photo sender retains control over who can view which parts of a privacy protected photo. And finally, the protected photo can be freely shared and viewed, albeit only in its protected form unless a requester has been authorized by the photo sender.

Figure 6.2 illustrates the overall architecture of ProShare, which consists of two types of components: client-side components and server-side components. Client-side components mainly refer to users local devices such as phone and laptop (including applications and software) on which the secure protection/recovery of images and the generation/encryption/decryption of secret keys take place. Users secret information, including the PKC private keys and ABE master secret keys, are kept on the client side. All client-side components are assumed to be trustworthy. ProShare users store their image data (protected)

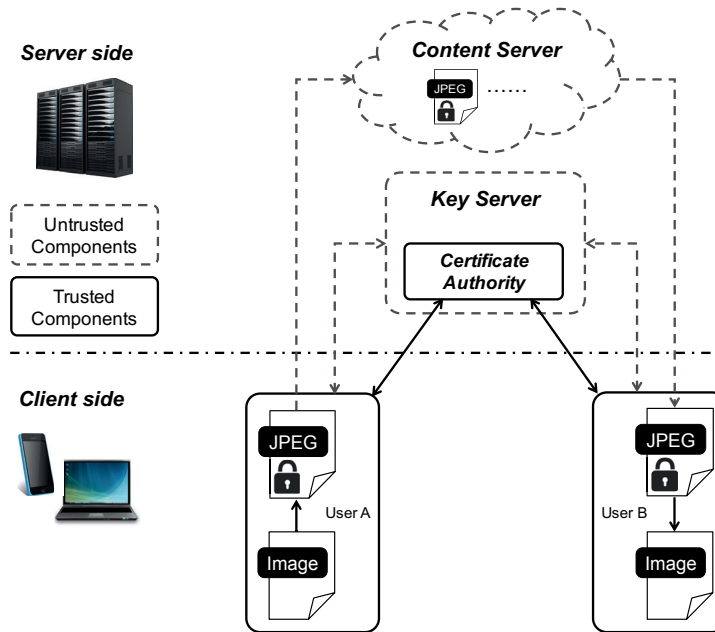


Figure 6.2 – Overview of ProShare architecture for privacy-preserving photo sharing.

including encrypted keys on a storage service (SS). Two types of servers are employed in each user’s storage service: (i) content server (CS), which is a specific server, storage space or cloud server (e.g., Dropbox) storing all users protected images; (ii) key server (KS), which keeps all the encrypted keys (image secret keys and ABE private keys). In addition, each ProShare user is identified by a user ID and a single PKC public key. We assume the existence of a Certificate Authority (CA) on the key server responsible for issuing public key certificate upon users requests. The content server and key server do not need to be trusted, but are assumed to operate correctly according to the proposed protocol, namely “honest-but-curious”. In addition, we assume all users do not permanently cache images they have viewed, image secret keys and ABE private keys on their client-side devices. The aim is to release the storage burden of the client side as much as possible, and more importantly for the purpose of revocation, which will be discussed in Section 6.2.2.

### 6.2.2 Operations

The complete flow of ProShare operations including how different cryptographic protocols collaborate is described in the following.

#### Initialization

First of all, a user registers an account by setting a username and password. Then the system (client-side) assigns the user an ID and generates two pairs of keys for the user:

---

**Function 1**  $\text{UserInit}(\text{Username}, \text{Password})$

---

- 1:  $i \leftarrow \text{GenerateUserID}(\text{Username}, \text{Password})$
  - 2:  $K_i \leftarrow \text{PBKDF2}(\text{Password})$
  - 3:  $(\text{TPK}_i, \text{TSK}_i) \leftarrow \text{PKCSetup}()$
  - 4:  $(\text{APK}_i, \text{AMSK}_i) \leftarrow \text{ABESetup}()$
  - 5:  $u_i.\text{SS.KS.put}(\text{TPK}_i)$
  - 6:  $u_i.\text{SS.KS.put}(\text{APK}_i)$
- 

---

**Function 2**  $\text{AddFriend}(u_i, \mathbb{A}_{i \rightarrow j})$

---

- 1:  $\text{ASK}_{i \rightarrow j} \leftarrow \text{ABEKeyGen}(\mathbb{A}_{i \rightarrow j}, \text{AMSK}_i)$
  - 2:  $C_{\text{ASK}_{i \rightarrow j}} \leftarrow \text{PKCEnc}(\text{ASK}_{i \rightarrow j}, \text{TPK}_j)$
  - 3:  $u_i.\text{SS.KS.put}(C_{\text{ASK}_{i \rightarrow j}})$
- 

PKC key pair  $(\text{TPK}, \text{TSK})$  and ABE key pair  $(\text{APK}, \text{AMSK})$ . In addition, a master key  $K$  for the user is generated by a password-based key derivation function (e.g. PBKDF2 [84]). The  $\text{TPK}$  is uploaded and managed by the centralized CA and the  $\text{APK}$  in the key server of the user. While, the user keeps his/her  $\text{TSK}$ ,  $\text{AMSK}$ ,  $K$  and a copy of the  $\text{APK}$  securely on the client side. The entire procedure is presented in Function 1.

### Friending

The user then invokes the Function 2 to add another user as friend featured by a set of attributes. The user generates an attribute secret key using the  $\text{ABEKeyGen}()$  function, encrypts this key using the public key of the target user, and then stores the encrypted key on the key server of his/her storage service.

**Example Usage:** Alice attempts to add Bob as friend and therefore generates Bob an ABE private key  $\text{ASK}_{\text{Alice} \rightarrow \text{Bob}}$  with the set of attributes: ( $\text{'Bob'}$ ,  $\text{'Close Friend'}$ ,  $\text{'Co-worker'}$ ). Alice encrypts this key using Bob's PKC public key  $\text{TPK}_{\text{Bob}}$  and stores the encrypted key  $C_{\text{ASK}_{\text{Alice} \rightarrow \text{Bob}}}$  on her storage service under the key server. Another user Carol, as a classmate of Alice, hopes to be a friend of Alice and therefore sends a request to Alice. Alice accepts the request and issues Carol an ABE private key in the same way as is for Bob, with a different set of attributes: ( $\text{'Carol'}$ ,  $\text{'Co-worker'}$ ,  $\text{'Education'}$ ).

### Photo Protection and Sharing

In this step, described by Function 3, the user protects a photo with a Secure JPEG protection algorithm (Scrambling or Transmorphing). The secret key for image protection is derived from the master secret key of the user, and is then encrypted with CP-ABE under an access structure  $\mathbb{S}$ , defined by the user in either manual or automatic way. The encrypted image secret key is uploaded to the user's storage service under the key server.

---

**Function 3** SharePhoto( $u_i, I, \mathbb{P}, \mathbb{S}$ )

---

```

1: salt ← RandomGen()
2: key ← PBKDF2( $K_i, salt$ )
3:  $I_P$  ← JPEGSec( $I, key, \mathbb{P}$ )
4:  $C_{key}$  ← ABEEnc( $key, \mathbb{S}, APK_i$ )
5:  $u_i.SS.CS.put([I_P, salt])$ 
6:  $u_i.SS.KS.put(C_{key})$ 

```

---



---

**Function 4** AccessPhoto( $u_j, I_P, u_i$ )

---

```

1:  $C_{ASK_{i \rightarrow j}}$  ←  $u_i.SS.KS.get()$ 
2: if  $C_{ASK_{i \rightarrow j}}$  exists then
3:    $ASK_{i \rightarrow j}$  ← PKCDec( $C_{ASK_{i \rightarrow j}}, TSK_j$ )
4:    $APK_i$  ←  $u_i.SS.KS.get(APK \text{ of } u_i)$ 
5:    $C_{key}$  ←  $u_i.SS.KS.get(\text{encrypted secret key for } I_P)$ 
6:   ( $result, key$ ) ← ABEDec( $C_{key}, ASK_{i \rightarrow j}, APK_i$ )
7:   if  $result$  is True then           # CP-ABE decryption is succeeded and  $key$  is valid
8:      $I'_O$  ← JPEGRec( $I_P, key$ )
9:     return  $I'_O$ 
10:  else                               # CP-ABE decryption failed and  $key$  is invalid
11:    Exit with warning: "Sorry, you have no right to access the original photo."
12: else                                 #  $C_{ASK_{i \rightarrow j}}$  does not exist
13:   Exit with warning: "Sorry, you are not a friend of the user."

```

---

**Example Usage:** Alice takes a photo with Carol and attempts to share this photo to Carol, to family and intimate friends. However, for some reason, she does not wish to share the entire photo Bob and others. So Alice protects an image region in the photo, uploads the photo to the photo sharing service (the content server) and sets the following access structure: ( ‘Family’ OR ‘Close Friend’ OR ‘Carol’ AND (NOT ‘Bob’) ). The secret key is then encrypted in CP-ABE with respect to the access structure.

### Photo Accessing and Viewing

On the other side, a requester  $u_j$  tries to recover and view the original version of a photo shared by  $u_i$ . The user firstly requests from the key server his ABE private key issued by  $u_i$ . If this key does not exist, it means  $u_j$  is not even a friend of  $u_i$  and therefore has no right to access the original picture. If the key exists,  $u_j$  decrypts the key using his PKC private key, and then uses the decrypted ABE private key to decrypt the image secret key in CP-ABE. If CP-ABE decryption succeeds, meaning that  $u_j$  has been granted the right to access the original photo,  $u_j$  recovers the image using the decrypted image secret key. When the complete photo viewing activity is finished and  $u_j$  has no long been watching the picture for a certain duration  $t$ , the decrypted image key and recovered image is deleted permanently from  $u_j$ 's device. The above operations are defined in Function 4.

**Example Usage:** Carol attempts to view the photo shared by Alice in the previous example. Since Carol has an attribute ‘Carol’ in her ABE private key issued by Alice, which satisfies the access policy defined for the photo: (‘Family’ OR ‘Close Friend’ OR ‘Carol’ AND (NOT ‘Bob’)), Carol is able to view the original photo. Bob also attempts to view the same picture but failed, although he is defined as a ‘Close Friend’ of Alice. This is because Bob is excluded by the access structure. Except for Bob, any other user in the group ‘Family’ or ‘Close Friend’ are able to see the original version of this photo.

### Revocations

Users connections in the environment of social networks are dynamic and may change over time. Users may also hope to change the access policy of the photos they shared in the past. Therefore, an efficient solution to revoke a user’s access right or to change a photo’s access policy is indispensable. In the current design of the proposed ProShare architecture, we hold the assumption that users would not permanently cache the image they have viewed, the image secret keys and ABE private keys they have decrypted on their client device. This means each time when attempting to view a photo of another user, the requester must execute the Function 4, namely, making a new request to the photo sharing service, retrieves and decrypts again corresponding keys from server. With this assumption, we provide two simple and flexible solutions for the two revocation problems respectively based on key re-encryption.

**Revocation of access right of a user:** To revoke or update the access rights of an existing friend, one can simply generate and encrypt a new ABE private key for that friend using a newly defined attribute set. Another extreme case would be to simply “unfriend” a user by removing the ABE private key of that user from key server. These operations are described in Function 5 and 6.

**Revocation of access policy of a photo:** This is the scenario where a user hopes to change the access range of her/his photos. In this case, the user can simply re-encrypt the image secret key, using a newly defined access policy. For example, to restrict the access of a photo that was available to (‘Family’ OR ‘Close Friend’ OR ‘Carol’), one can re-encrypt the image key using the new access policy ( ‘Family’ OR ‘Carol’ ), to make it only accessible to family members and Carol. Another particular case is to delete a photo, where one can simply remove the photo and corresponding secret keys from server. This process is presented in Function 7 and 8.

Alternatively, the above approaches based on key re-encryption for revocation provide users with flexible solutions to grant more access rights to friends or increase access range of photos. Since the generation of ABE private keys and CP-ABE encryption operations are proven fast enough with a limited number of attributes, e.g., below 20 [82], such

---

**Function 5** UpdateFriend( $u_i, \mathbb{A}_{i \rightarrow j}^*$ )

---

/\*  $\mathbb{A}_{i \rightarrow j}^*$  is a new set of attributes. \*/

- 1:  $u_i.SS.KS.remove(C_{ASK_{i \rightarrow j}})$
  - 2: AddFriend( $u_i, \mathbb{A}_{i \rightarrow j}^*$ )
- 

---

**Function 6** RemoveFriend( $u_i$ )

---

- 1:  $u_i.SS.KS.remove(C_{ASK_{i \rightarrow j}})$
- 

revocation approaches are efficient and flexible in most practical cases. More advanced approaches to revocations [27, 85] can also be used, both relying on a minimally trusted proxy to handle revoked users and attributes.

### Negation in Access Structure

Negation expression like (NOT ‘Bob’) is not directly supported in the common implementation of CP-ABE [82]. However, some negation expressions can be solved by being converted to numerical expressions, making use of the unique ID of user. Let us assume the ID is a number, for example, Bob holds the ID of 8. The negation expression (NOT ‘Bob’) in above example could be converted to (‘UserID < 8’ OR ‘UserID > 8’). Similarly, conjunction of negation expressions can be interpreted as combination of several numerical comparison expressions, e.g. ((NOT ‘Bob’) AND (NOT ‘David’)) is equivalent to ((‘UserID < 8’) OR (‘UserID > 8’ AND ‘UserID < 16’) OR (‘UserID > 16’)), where 16 is the ID of David.

## 6.3 Prototype and Implementation

A prototype application named ProShare has been developed to demonstrate the minimum functionalities of the proposed photo sharing architecture. We use RSA (key length 1024-bit) in PHP<sup>1</sup> as the public key encryption algorithm and the `cpabe` toolkit<sup>2</sup> to implement the functionalities of CP-ABE. For Secure JPEG image protection, we demonstrate JPEG Transmorphism with scrambling as the encryption scheme for securing sub-image.

The prototype application consists of two parts: (i) a client mobile interface, running on both iOS and Android platforms, and (ii) a web server hosting images and managing secret keys. For the ease of implementation, the protection of images, secret key management and various encryption and decryption operations all perform on the server. Instead, the mobile application simply acts as a user interface. In such a way, the implemented prototype application is designed to simulate the behavior of the proposed photo sharing

---

<sup>1</sup><http://php.net/manual/en/function.openssl-public-encrypt.php>

<sup>2</sup><http://hms.isi.jhu.edu/acsc/cpabe/>



---

**Function 7** UpdatePhoto( $u_i, I_P, \mathbb{S}^*$ )
 

---

/\*  $\mathbb{S}^*$  is a new access structure. \*/

- 1:  $salt \leftarrow u_i.SS.CS.get(\text{the salt of } I_P)$
  - 2:  $key^* \leftarrow \text{PBKDF2}(K_i, salt)$  #  $key^*$ : a new image secret key
  - 3:  $C_{key}^* \leftarrow \text{ABEEnc}(key, \mathbb{S}^*, \text{APK}_i)$  #  $C_{key}^*$ : re-encrypted image secret key
  - 4:  $u_i.SS.KS.replace(C_{key}, C_{key}^*)$
- 

---

**Function 8** RemovePhoto( $u_i, I_P$ )
 

---

- 1:  $u_i.SS.KS.remove(C_{key})$
  - 2:  $u_i.SS.CS.remove([I_P, salt])$
- 

architecture. In our implementation, the web service consists of three components, each for performing/storing a set of ProShare operations/data:

- **Trusted Component (TC):** hosts users certain secret information such as PKC private keys and ABE master keys. The Secure JPEG protection and recovery also happen on this component. We use this part of the web service to simulate some of encryption/decryption operations supposed to happen on client device, which is assumed to be trusted.
- **Key Service Component (KSC):** acts as the key server defined in Section 6.2, for storing encrypted keys and issuing PKC public keys.
- **Content Service Component (CSC):** acts as the content server defined in Section 6.2, for storing users protected images.

#### 6.3.1 Functionalities

The prototype application simulates the following minimum functionalities of the proposed photo sharing architecture.

**User Registration** Each user registers an account using an e-mail address. Upon registration, two pairs of keys (TPK/TSK, APK/AMSK) are generated. The two public keys (TPK and APK) are then stored and managed by KSC and the other two secret keys are stored to on TC in each user’s storage space<sup>3</sup>. This step happens on the TC.

**Photo Protection and Sharing** User takes a picture from the mobile device and applies a Transmorphic protection by placing one or more image stickers (selected by

---

<sup>3</sup>Each user has an individual folder on TC.



Figure 6.3 – Screenshots of ProShare iOS application.

user) onto sensitive image regions. The mask matrix is computed on the mobile device and then sent to TC along with the original image, protected image, image secret key and defined access policy. Then the actual protection based on Secure JPEG Transmorphing takes place on TC. Once finishing protection, the secure photo is uploaded and stored on the CSC while the original image and mask matrix are deleted. Meanwhile, the image secret key is encrypted in CP-ABE under the user-defined access policy. In the current implementation, the image secret key is manually set by user.

**Friendship Management** The application allows user to add friend by specifying the email address of another user and a set of attributes. In the current implementation, we only consider a certain number of relation-based attributes, such as ‘Family’, ‘Close Friend’, ‘Colleague’, ‘Education’ and ‘Current city’. The friend’s ASK is then generated and encrypted according to the algorithm described in Section 6.2. Both the generation and PKC encryption of ASKs happen TC.

**Photo Viewing** In the prototype application, all photos are by default public to everyone in protected form. Only those who were assigned the authorized attributes are able to view the original photos. The key decryption and image reconstruction processes take place on TC. The recovered image is sent to and displayed on the mobile phone. After finishing reviewing the photo, the recovered image along with decrypted image keys are deleted from TC.

**Interaction with Facebook** In addition, the application also allows user to share images protected in Secure JPEG on Facebook in the form of App Links<sup>4</sup>. By clicking the

<sup>4</sup><http://applinks.org/>

shared link, other Facebook users are directed to either the ProShare App (if ProShare installed on the mobile device) or a web interface (otherwise). If the user has an account on ProShare, the photo can be shown to the user depending on his/her “relation” (attributes) with the photo sender. On the web interface, a secret key has to be manually provided in order to recover the image.

Two interfaces (iOS and Android) of the prototype application have been developed and made publicly available on the Apple Store<sup>5</sup> and Google Play<sup>6</sup> respectively. Screenshots of the iOS App demonstrating ProShare are shown in Figure 6.3.

### 6.3.2 Evaluation

Based on the prototype implementation, we conducted a simple evaluation experiment to examine the time required for key ProShare operations: (i) adding friend (Function 2), (ii) share photo (Function 3) and (iii) access photo (Function 4). The evaluation was performed using an iPhone 5C as interface and all images were taken from the iPhone camera and resized to a smaller size (max. width or height of 1080 pixels) before being uploaded to the server. The server employed has an 8-core Intel(R) Core(TM) i7 CPU with 2.80GHz and 16GB memory. To evaluate the time for adding friend, we used the application to add 8 target users as friend using different number of attributes, from 1 to 8, respectively. To evaluate the time for sharing photos, we selected 30 photos from iPhone gallery (taken either from frontal or back camera), apply protection using a random cartoon sticker on a face region, and share each photo with CP-ABE using different policies containing varying numbers of attributes with conjunction “OR”. The number of attributes is changed from 1 to 8. To evaluate the time for accessing photo, we used each of the 8 target users’ account to access each of the 30 photos protected the 8-attributes policy. Here we assume the photo accessing time may depends on the number of matched attributes between the user’s ABE private key and the access policy.

Figure 6.4 displays measurements (mean and 95% confidence interval) of time for adding friend, sharing photo and accessing photo respectively. From the result, one observes that the time for photo sharing and friend addition operations linearly increases as the number of attributes increases, and this result well agrees with the performance results of the `cpabe` toolkit in [82]. Even with 8 attributes used, it only takes no more than 0.18 seconds in average to share a photo, which are short enough so that one can hardly feel. Adding friends takes even less time than sharing photos. For accessing photos, the time slightly increases as the number of “matched” attributes increases. Such an increasing trend is not obvious and the time spent is always in range of 0.06 and 0.08 seconds. Notice that all these operations also include the photo uploading, PKC encryption/decryption, and JPEG Transmorphing protection/recovery; in this respect, our prototype application

---

<sup>5</sup><https://itunes.apple.com/us/app/proshare/id1047578277>

<sup>6</sup><https://play.google.com/store/apps/details?id=ch.epfl.proshare>

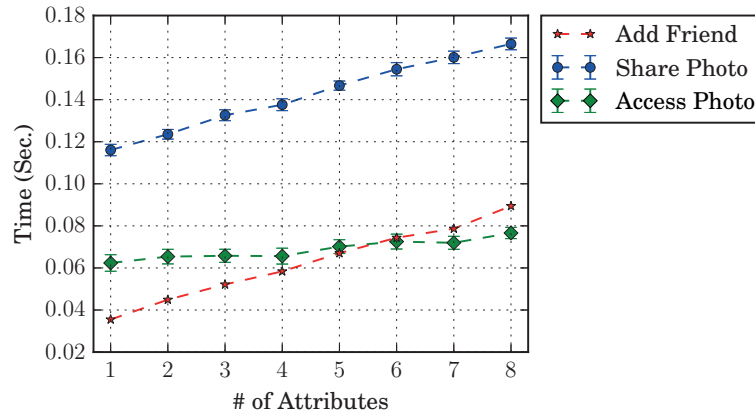


Figure 6.4 – Performance evaluation of ProShare prototype application.

demonstrates an efficient and fast functioning of the ProShare photo sharing architecture.

## 6.4 Conclusion

In this chapter, we present ProShare, an architecture for privacy-preserving photo sharing based on a public key infrastructure. In ProShare, a sender can select and protect an arbitrary set of regions in a given photo, using any preferred visual obfuscation, powered by the Secure JPEG privacy protection algorithms. Once protected, the resulting image file can be freely shared and viewed with any JPEG compliant decoder. The shared image file is internally consistent and therefore contains all the data necessary to view both protected and unprotected image regions. Finally, the photo sender can dynamically associate an access policy with the protected photo and assign a set of attributes with prospective requester. Only if the photo sender has granted the requester a set of matching attribute(s) compared to the access policy defined for the photo, will this photo be accessible to that particular requester. Complying with the above attributes, such a service allows for the protection of privacy in photos prior to them leaving the user's device. Once shared, the photo sender retains control over who can view which parts of a privacy protected photo. And finally, the protected photo can be freely shared and viewed, albeit only in its protected form unless a requester has been authorized by the photo sender. In the end, we implemented a prototype application built on a dedicated server communicating with two mobile interfaces, i.e. iOS and Android, to demonstrate the correct and efficient functioning of the proposed architecture.

## 7 ProShare S: Context-Dependent Privacy-Aware Photo Sharing

Most social networking or photo sharing services provide access control for users to manage who can access their photos. However, in most services, users need to manually set their policies in a static manner, without the possibility to share their photos to different groups of people depending on contexts, e.g. the location, time or even nearby people of the prospective viewer. Most access control mechanisms enforce only binary sharing options, namely “Yes” or “No”, which may not provide the best experience when a user just wants to disable partial information in photo sharing. With the recent advancements in image analytics, pattern recognition, and deep learning techniques, large scale information is mined from multimedia content shared by users. These information is used for different purposes by service providers, such as content and advertisement recommendations. Those techniques, though seemingly compromising privacy, can in turn be used to enhance our privacy, in such a way of helping people estimate the privacy value of their content or control the access of their content automatically and dynamically.

In this chapter, we present a conceptual architecture for photo sharing, named ProShare S, where the service provider is granted the trust to help users make photo sharing decisions automatically based on their past decisions. The proposed architecture takes into account not only the content of a user-posted image, but also the context information about the image capture and a prospective requester. Using machine learning, the system then makes decision whether or not to share a particular photo of a user to a requester, and if yes, at which granularity.

The rest of the chapter is structured as follows. Section 7.1 describes in detail the proposed architecture for context-dependent and privacy-preserving photo sharing. Then Section 7.2 and Section 7.3 present the user study and corresponding evaluations performed on the collected dataset. Finally, Section 7.4 outlines some discussions and Section 7.5 concludes this chapter.

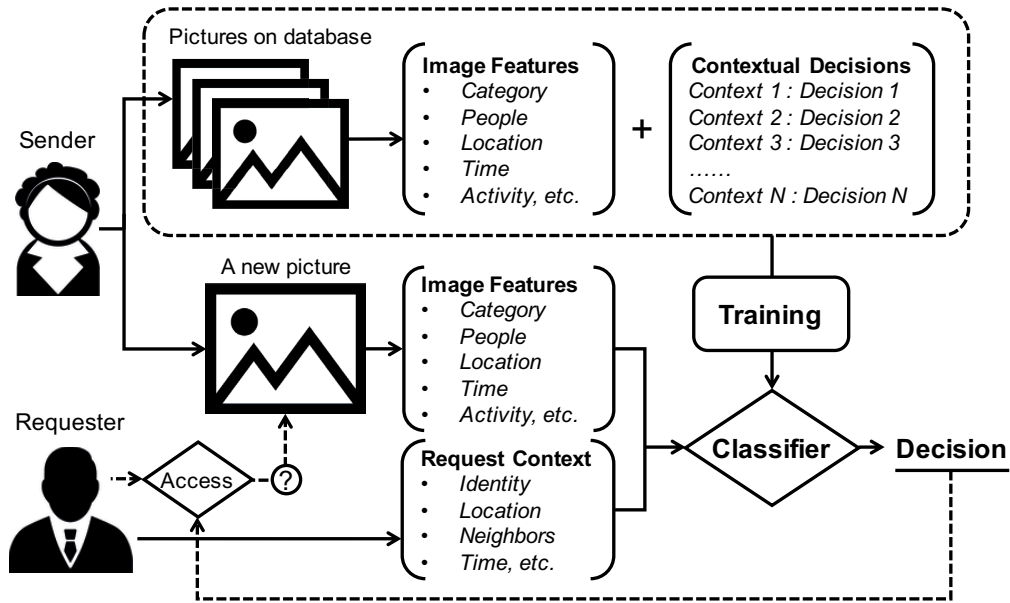


Figure 7.1 – Workflow of ProShare S architecture.

## 7.1 ProShare S: The Architecture Design

ProShare S is a “brand new” architecture design for privacy-preserving photo sharing that employs different philosophy compared to the ProShare in Chapter 6. In ProShare S architecture, the photo sharing service is minimally trusted to apply necessary image analysis and pattern recognition techniques on user posted photos. However, the information extracted from users photos is used to understand the privacy value of those photos, and then to decide whether or not a particular photo can be disclosed to another user. The proposed architecture utilizes not only the semantic features of photos but also the contextual features of the photo capture and the requester, to build a photo sharing decision making core based on machine learning. The architecture is illustrated by an example in Figure 7.1. In the following, we describe the operating principle and the architecture design in detail.

### 7.1.1 Operating Principle

#### Security Assumption

First of all, we assume the photo sharing service providers are trustworthy. This means users allow the service to conduct necessary analysis on their photos, and the system is granted the right to enforce access control of users photos. Therefore, the proposed architecture is mainly to guard users privacy against the spying from unauthorized, curious or malicious people. Actually, it is possible to relax the security assumption of the ProShare S architecture, in given conditions. This will be discussed in Section 7.4.

### A Practical Scenario

First we assume Alice is the sender and Bob is a requester. The definitions of the sender and requester can be found in Section 6.2. We then use the following story to describe the operating procedures of the proposed architecture: Alice uploads a set of pictures on the photo sharing service, and the service system analyzes each picture and extracts a set content and contextual features about those images. Meanwhile, the system asks Alice a set of questions on her willingness to share each picture to specified individuals in various scenarios. These individuals can be selected from those who visited Alice’s profile recently or frequently. Each scenario describes a certain context of a possible requester, who attempts to visualize a picture shared by the sender. The context includes the identity (either real name or social group), location, nearby people and the time when the requester tries to visualize the image. The system then trains a classifier based on Alice’s answers for different photos in different scenarios. On the other side, Bob visits the profile page of Alice. With the help of the classifier, the system analyzes Bob’s context and Alice’s photo information, to decide whether or not to show certain photos to Bob, and if yes, at which granularity.

#### 7.1.2 Feature Definition

To train such a classifier, we considered two groups of features: **Image Semantic Features (I)** and **Requester Contextual Features (R)**. Instead of low-level image features such as color, texture, composition and SIFT, we considered higher-level semantic features which we believe have a more immediate influence on sharing decision. These features include the image category, number/identities of people in image, activities or objects in image and the location and time of image capture. The contextual features of the requester include the requester’s identity, location, nearby people and time.

A detailed description of all the features used in our experiments, grouped in different aspects of context, is shown in Table 7.1. Note that the time of requester is not included in this list, because it would be too cumbersome for subjects to read and analyze the complete information containing all contexts. Also, in our study, we could not put subjects in realistic photo sharing scenarios, due to the lack of a real social networking system. Therefore, we used only social groups to define the identities of requester, and added the gender of requester as another feature. According to different natures of features, they can be defined in different data types, such as numerical or categorical. Particularly, we applied a simplified Bag-of-words model to describe the people identities and activities in image, because more than one identities or activities can be in the same image.

Table 7.1 – Notation and definition of features in ProShare S.

	ID	Feature	Description
What	$I_C$	Image: Category	Major category of the picture, selected from the eight categories identified in Instagram pictures [86]: <i>Friends, Activity, Selfie, Food, Pets, Gadget, Fashion</i> and <i>Captioned photo</i> .
	$I_A$	Image: Activities	Activities involved in the picture, selected from 26 keywords partially defined by [87]: <i>working, meeting, reading, presentation, resting, chatting, socializing, family, friends, vacation, TV, cooking, eating, drinking, cleaning, shopping, exercising, traveling, walking, landscape, city, concert, sporting, gaming, gadget</i> and <i>pets</i> .
Who	$I_P$	Image: # of People	The number of people in the picture.
		Image: Identities	The existence of different identities in the picture. Eight types of identities were defined: <i>Sender him/herself, Family, Close friend, Schoolmate or Colleague, Girl or Boyfriend, Acquaintance, Celebrity</i> and <i>Stranger</i> .
	$R_I$	Requester: Identity	The relationship between the requester and the sender, categorized in six types: <i>Family, Close friend, Schoolmate or Colleague, Girl or Boyfriend, Acquaintance</i> and <i>Stranger</i> .
	$R_G$	Requester: Gender	Gender of the requester: <i>Female</i> or <i>Male</i> .
	$R_N$	Requester: Nearby	Whether or not the requester has other people nearby at requesting time.
Where	$I_L$	Image: Location	The semantic location where the image was captured, selected from 12 major location categories adopted from Foursquare Location Categories.
		Image: Loc. Coordinates	Latitude and longitude of the image capture location.
		Image: Loc. Frequency	The frequency of the sender being present in such place, selected from <i>Rarely, Sometimes, Often</i> and <i>Almost everyday</i> .
	$R_L$	Requester: Location	Semantic location of the requester, categorized in <i>Unknown, Friend's home, His/her own home, Work place</i> and <i>Public place</i> .
When	$I_T$	Image: Time	The time of photo capture in a float value, e.g. 14.5 denotes 2:30 PM.
		Image: Day	The day (in a week) of photo capture, selected from <i>Monday to Sunday</i> .



### 7.1.3 Photo Sharing Decisions

In the proposed photo sharing architecture, we define three sharing decisions, corresponding to different levels of photo information disclosure. The three decisions and corresponding descriptions presented in the user study are as follows:

**Decision 1 - Do NOT Share:** No, I don't want to share the picture.

**Decision 2 - Partially Share:** Yes, but with some image region protected or/and metadata (GPS, time, etc.) removed.

**Decision 3 - Entirely Share:** Yes, I want to share the picture completely.

The reasons of using the specific three sharing decisions instead of conventional binary decisions ("Yes" or "No") are twofold: First, in many scenarios of online photo sharing, people may want to simply remove partial privacy-sensitive visual information in an image, such as ID card, license plate or their children faces. Second, most images shared from smart mobile devices contain metadata such as geotags, camera model and time, which could also compromise privacy. Therefore, an option should be provided for users to partially protect and share their image content. Within the framework of ProShare S, Secure JPEG protection described in Part I of the thesis can be just used to preserve partial image privacy corresponding to the second decision. To do so, the system can create a secure version of a photo, with protect region defined by user. Depending on the predicted decision, the system then releases the corresponding version (protected or recovered original form) of the image to the requester.

## 7.2 User Study and Data Collection

We conducted a study that put participants in personalized photo sharing scenarios, and collected an image dataset containing user-annotated image semantic features and personal contextual sharing decisions.

### 7.2.1 The Data Collector

To conduct user study and collect data, we developed an Android application<sup>1</sup>, named *ProShare S*. Several screenshots of the application are shown in Figure A.6 in Appendix A. The application allows a user to create an account, take pictures, conduct a set of surveys for each, protect privacy-sensitive image regions, and finally upload them to a dedicated server. The workflow of a user study using ProShare S is illustrated in Figure 7.2. Particularly, the survey part is structured in two sets of questionnaires:

---

<sup>1</sup>The application is publicly available at <http://grebvm2.epfl.ch/proshare-s/proShare-rd2.1.apk>.

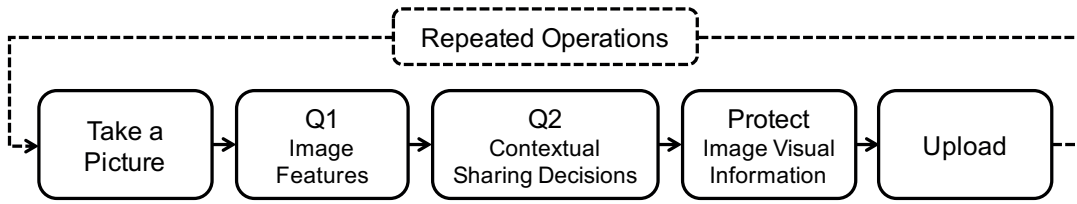


Figure 7.2 – Workflow of user study using ProShare S.

**Q1 - Image Semantic Information** The first questionnaire (Q1) requires the user to add necessary image semantic tags. This questionnaire appears once a picture has been taken from either gallery or camera. The questions in Q1 cover all the semantic features defined in Section 7.1.2. A build-in face detector offered by Android API<sup>2</sup> is applied to count the number of people in image, which can be manually modified if not correct. Location coordinates and capture time are automatically extracted from image metadata.

**Q2 - Contextual Photo Sharing Decisions** Once Q1 is finished, the user is directed to the second questionnaire (Q2), where he/she is presented with 12 sharing contexts/questions. For each context, user needs to decide how he/she would like to share the picture with the specific requester, by selecting one of the three decisions defined in Section 7.1. The description of an example context is “Would you share this picture with a *close friend*, when *he* is at a *public place* with *other people*?” The 12 contexts (or questions) are selected in a special way such that each of the six requester identities appears twice in a random order, with the other contextual features (gender, location, nearby people) sampled at random. In the study, basic user profile is also collected through the app. We therefore present the sharing contexts adaptively based on user’s profile. For instance, for a female user we present the requester as “your boyfriend” instead of “girl or boyfriend”.

## 7.2.2 User Study and Dataset Basic Statistics

We recruited 23 volunteers to participate in our user study, and assigned each of them a task of uploading at least 50 daily pictures of their own and completing corresponding surveys using ProShare S. Each subject was required to complete the task within a week and was asked to try to cover a wide range of image content<sup>3</sup>. Finally, 20 out of the 23 subjects successfully finished the required task. We therefore kept only the data of the 20 effective subjects for the later evaluation experiment. A total of 1’018 images including 12’216 sharing decisions were contributed by the 20 subjects, each providing 50.9 images on average<sup>4</sup>. Figure 7.3 shows the histogram of images in each content category

<sup>2</sup><https://developer.android.com/reference/android/media/FaceDetector.Face.html>

<sup>3</sup>The instruction and agreement sheet for the user study is available at [http://grebvm2.epfl.ch/proshare-s/instruction\\_sheet\\_rd2.1.pdf](http://grebvm2.epfl.ch/proshare-s/instruction_sheet_rd2.1.pdf)

<sup>4</sup>The dataset containing all image semantic features and users sharing decisions is publicly available at [http://grebvm2.epfl.ch/lin/thesis/dataset/data\\_ProShareS.zip](http://grebvm2.epfl.ch/lin/thesis/dataset/data_ProShareS.zip)

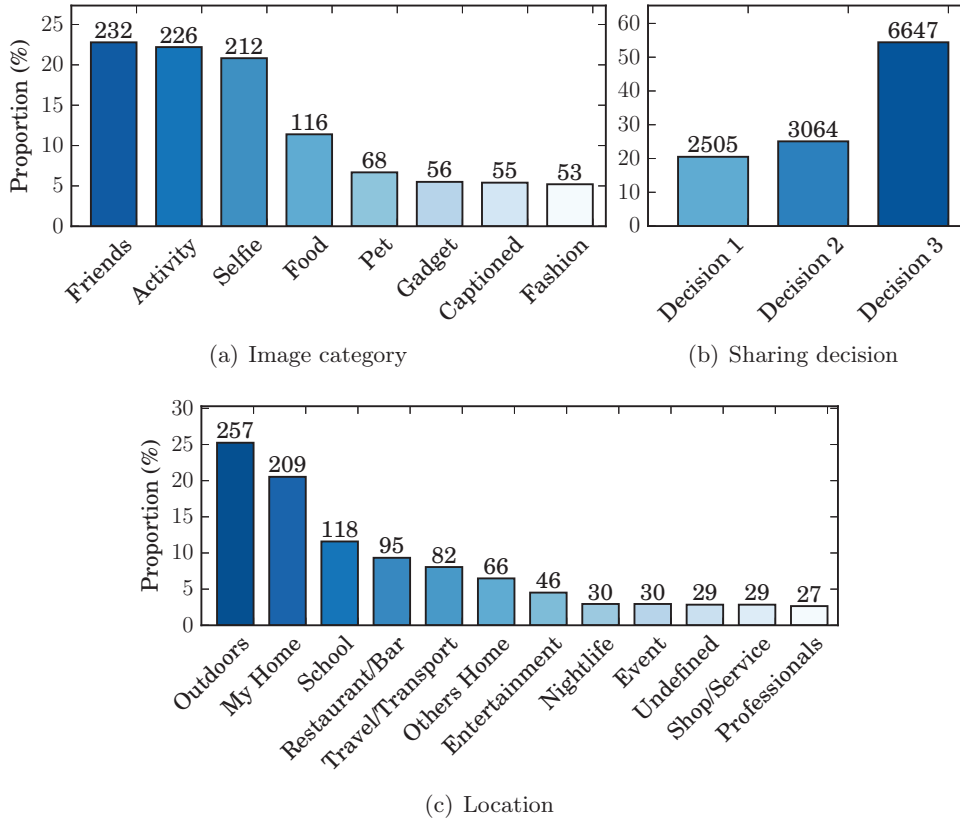


Figure 7.3 – Distribution of (a) images in each category, (b) subjects sharing decisions and (c) images in each location type.

and location type, and the contextual sharing decisions made on all the images. The distribution of images over different image categories (Figure 7.3(a)) basically agrees with the observations found in [86]. From Figure 7.3(b), users sharing decisions are more biased to Decision 3 (“Entirely Share”) instead of Decision 1 and 2. In addition, one observes a significant number of decisions made on Decision 2, slightly higher than Decision 1, which indicates the necessity of providing the third photo sharing option (“Partially Share”) for users instead of only binary sharing (“Yes” or “No”).

### 7.3 Evaluation and Analysis

To evaluate the performance of the proposed model for decision making, we conducted three sets of experiments based on the data collected from our user study. We take the working hypothesis that users photo sharing behaviors and privacy attitudes are highly subjective and such behaviors or attitudes may change over time from an image to another. The variance of user behaviors may also cause the proposed model to perform differently between subjects.

### 7.3.1 Methodology

The first experiment focused on the performance of the proposed model with respect to each user, namely, within-subject analysis. In the second experiment, we explored a universal one-size-fits-all classifier trained on all users data for predicting a new user's decisions. In the third experiment, we investigated the influences of different image and requester features on the decision making performance of the proposed model.

The WEKA machine learning library<sup>5</sup> [88] was used in experiments and three representative classification methods were considered: logistic regression, support vector machine (SVM) and random forest. We started with a preliminary test by running a 10-fold cross validation on each user's data using the three methods and random forest always outperformed the other two. We therefore kept using random forest for the rest of the experiment.

To evaluate the decision making performance of the proposed architecture, we use the following metrics:

- **Correct Decision rate:** The proportion of correctly predicted decisions.
- **Over-Sharing rate:** The proportion of cases where image information is shared more than what user expect to share, which compromises privacy.
- **Under-Sharing rate:** The proportion of cases where image information is shared less than what user expects to share, which may compromise usability.
- **Kappa statistic:** Cohen's kappa score [89] that measures the chance-corrected agreement between predicted and ground truth decisions.

### 7.3.2 Within-Subject Analysis

In the first experiment, we considered the scenario where users need firstly make a number of sharing decisions manually so that the service can train a classifier to make the remaining decisions automatically for users. This is to examine the trade-off between user-burden and prediction accuracy of the proposed model. In this experiment, we used different proportions (from 10% to 90%) of each subject's data to train a classifier, and evaluated the classifier on the rest of the data (evaluation set). The evaluation results measured by different metrics across the 20 subjects are shown as box plots in Figure 7.4. In this figure, one observes that the median correct decision rate has already reached 0.75 at a training set of only 10%, which corresponds to only 5 images in average. This means we could already build an acceptable model for half of the users using a very small number of images and their decisions. Above the training set of 50%, most users obtained the

<sup>5</sup><http://www.cs.waikato.ac.nz/ml/weka/>

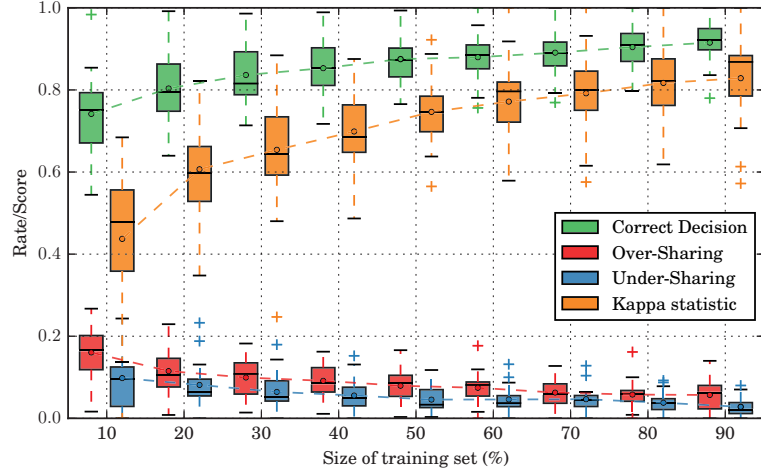


Figure 7.4 – Performance of decision making at different sizes of training sets.

correct decision rate higher than 0.8. The median Kappa score at the training set of 10% is below 0.5 and rapidly reaches 0.6 at the training set of 20%. Above the training size of 60%, an almost perfect prediction is observed for half of the users with a median Kappa statistic greater than 0.8. On the other hand, both the over-sharing and under-sharing rates of most users are very low, even at the training set of 10%. However, we observe the over-sharing rate is always higher than the under-sharing rate. A possible explanation is that most users tend to share images and the numbers of different decisions in the dataset are imbalanced. From the results, one also observes a significant variance between users. At the training size of 10%, the maximum difference in correct decision rate between users is up to 0.44. At the training size of 80%, where the optimal performance is obtained for most of the users, such difference still remains around 0.2. Such results agree with our hypothesis made in the beginning of this section that users subjective behaviors may influence the performance of the proposed model.

### Cost-Sensitive Learning

To address the issue of over-sharing, we introduced the cost-sensitive learning [90] in our decision making core. The aim is to evaluate the extent to which incorrect decisions can be biased towards the under-sharing cases instead of over-sharing, when users concern their privacy more than usability. We specified different error-penalties  $C$  ( $> 1$ ) for over-sharing cases and the same penalty of 1 for all under-sharing cases. Therefore, the training process tries to minimize the following cost equation:

$$\text{Total Cost} = \sum_{1 \leq i < j \leq 3} (C_{i \rightarrow j} \times N_{i \rightarrow j} + 1 \times N_{j \rightarrow i}), \quad (7.1)$$

where  $N_{i \rightarrow j}$  denotes the number of cases where Decision  $i$  is misclassified classified

Table 7.2 – The cost matrix applied in cost-sensitive learning.

↓ classified as →	Decision 1	Decision 2	Decision 3
Decision 1	0	$C_{1 \rightarrow 2} = c$	$C_{1 \rightarrow 3} = 2c$
Decision 2	1	0	$C_{2 \rightarrow 3} = c$
Decision 3	1	1	0

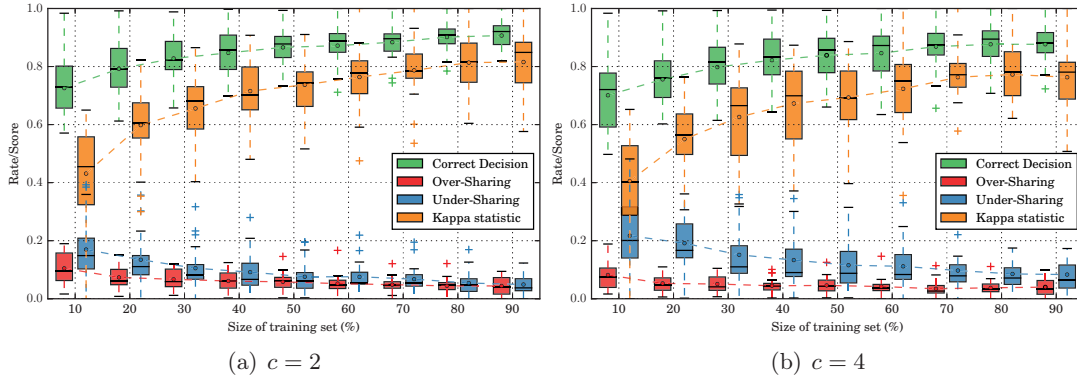


Figure 7.5 – Performance of cost-sensitive decision making with two different values of  $c$ .

as Decision  $j$ . Specially, we assigned a double error-penalty  $2c$  for the over-sharing cases  $C_{1 \rightarrow 3}$  compared to the other two over-sharing cases. This is because a mistake by classifying “Do NOT Share” to “Entirely Share” may severely compromise privacy. The cost matrix for the cost-sensitive learning is shown in Table 7.2.

We experimented with a set of values for  $c$  (from 1.5 to 5), on each user’s data using the same random forest classification. The results at  $c = 2$  and  $c = 4$  are shown in Figure 7.5. With an error-penalty  $c = 2$ , the over-sharing rate is greatly reduced to a level lower than the under-sharing rate. When increasing  $c$  to 4, the over-sharing rate is further reduced, in sacrifice of a significant increase on the under-sharing rate. This indicates a significant trade-off between the capability of privacy protection and system usability. In any cases of cost-sensitive learning, the overall correct decision rate and Kappa statistic do not change much, as the introduced error-penalty mainly acts as a parameter to tune the weights of different incorrect decisions.

### 7.3.3 One-Size-Fits-All Model

In the second experiment, we evaluated a one-size-fits-all model, to examine the potential of building a global classifier trained on the data of all users for predicting decisions on a new user’s images. Such a model could be useful when a new user has no enough images or decisions to build a personalized classifier, in which case the global classifier can help the user make or “recommend” decisions. In this experiment, for sake of fairness, for

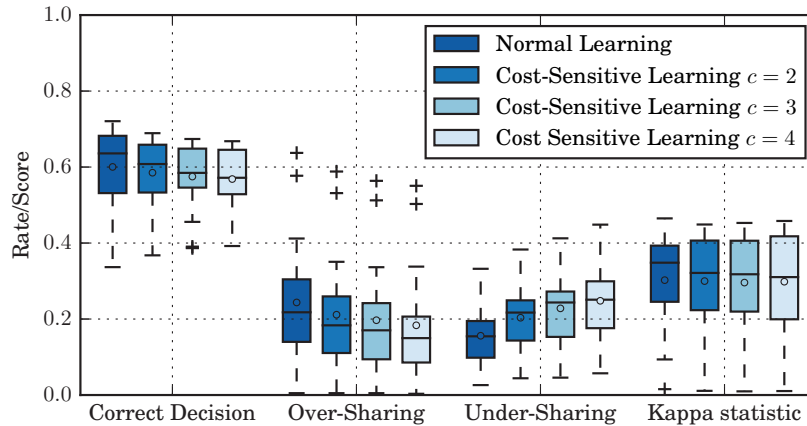


Figure 7.6 – Performance of a One-Size-Fits-All classifier on decision making.

each subject  $i$ , we trained a classifier using random forest on the data of all the *other* subjects. The classifier was then evaluated on the data of subject  $i$ . This corresponds to a training-to-evaluation ratio of 19:1 in average. Cost-sensitive learning was also included in this experiment for comparison. The results over all the 20 subjects are shown in Figure 7.6. The median correct decision, over-/under-sharing rates and the Kappa statistic without cost-sensitive learning are 0.636, 0.218, 0.155 and 0.348 respectively. With cost-sensitive learning, the over-sharing rates are reduced below under-sharing, without greatly degrading the correct decisions and Kappa score. The overall performance of such a one-size-fits-all model is not as good as the personalized classifier built on each user’s own data. This again implies that users may have very different behaviors and privacy attitudes towards photo sharing. Nevertheless, such a classifier could already provide a much more accurate prediction performance than a random guess, the correct decision rate and Kappa statistic of which would be about only 0.33 and 0.

### 7.3.4 Influences of Features on Decision Making

At the end, we investigated the influences of different types of features on users photo sharing decisions and on the performance of our prediction model. First, the histograms of three sharing decisions distinguished by different types of features are shown in Figure 7.7. The variation in decision distributions over different feature values indicates the influence degree of a particular feature type. One observes a significant difference in decision distributions across different identities of requester, which implies that the requester identity influences users decision making the most. On the other hand, although the decision distribution does not change much between other contextual features of the requester, there is still a small decrease of the “Entirely Share” decisions at the cases where the requester is at an “Unknown” place or with “Other people” nearby. Different types of image semantic features also have moderate degrees of influence on the sharing decisions. For instance, users tend to share pictures without people or with a lot of people

## Chapter 7. ProShare S: Context-Dependent Privacy-Aware Photo Sharing

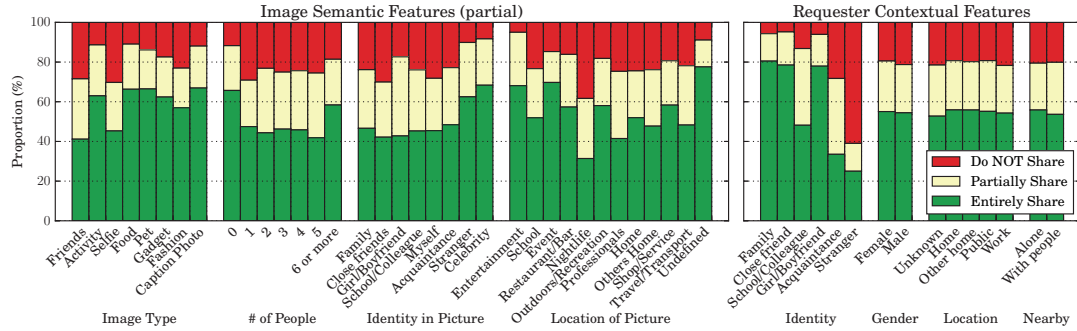


Figure 7.7 – Histogram of photo sharing decisions distinguished by different features.

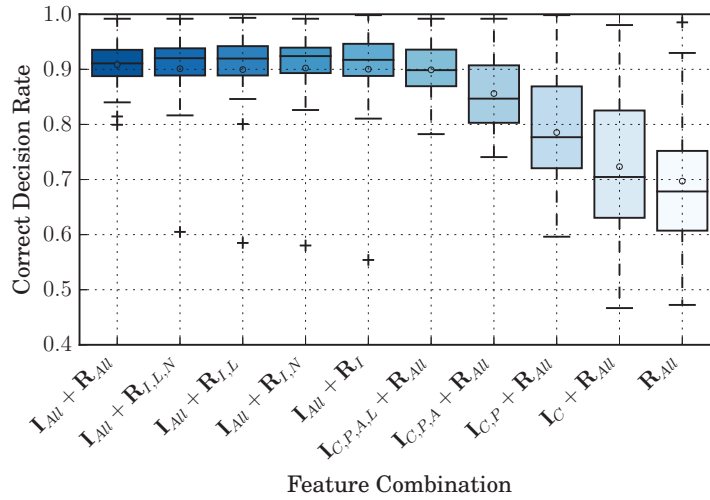


Figure 7.8 – Correct decision rates obtained on different combinations of features.

( $\geq 6$ ), more than the pictures with 1 ~ 5 people. Also, users favor sharing the pictures containing strangers or celebrities, over personal pictures with closer contacts like family and close friends.

We then evaluated performance of decision making on different combinations of image and requester features, by conducting a 10-fold cross validation on each user’s data. The correct decision rates of cross validation of all the 20 subjects are shown in Fig. 7.8. Please refer to the feature notations in Table 7.1. We gradually remove certain features, and the leftmost and rightmost box plots in Figure 7.8 show two extreme cases where all the features ( $I_{All} + R_{All}$ ) or only the requester features ( $R_{All}$ ) were used. As is shown, when reducing features, the correct decision rate of the majority of subjects decreases, which implies that all those features in general have a positive impact on decision making for most users. When reducing image-related features, a significant variance across different subjects is observed, which indicates that those image features are important for modeling many users sharing decisions. However, for two of those subjects, the prediction model



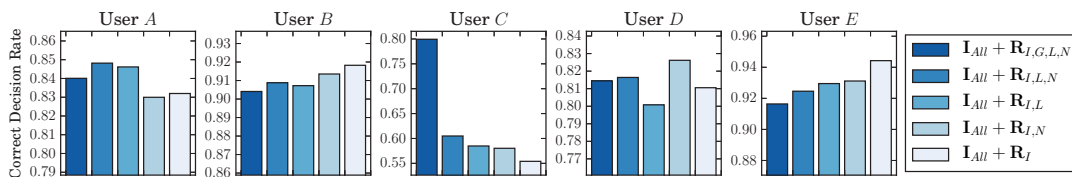


Figure 7.9 – Correct decision rates obtained on combinations of all Image Semantic Features ( $\mathbf{I}_{All}$ ) and different Requester Contextual Features ( $\mathbf{R}$ ) for five example users.

still performs well (correct decision rate higher than 0.9) even using only the requester features ( $\mathbf{R}_{All}$ ). A possible reason is that the two users made their sharing decisions mostly dependent on the context of requesters, regardless of the image content.

One also observes that by removing certain requester contextual features, such as requester gender ( $\mathbf{R}_G$ ), location ( $\mathbf{R}_L$ ), or nearby people ( $\mathbf{R}_N$ ), the overall accuracy does not significantly change. With merely the requester identity ( $\mathbf{R}_I$ ) + all image features ( $\mathbf{I}_{All}$ ), the overall decision making accuracy still remains high. This implies that the requester contextual information than the requester identity has very weak or even negative influence on decision making. However, this is not always the case for every subject. Fig. 7.9 illustrates the results of five example subjects obtained on different combinations of requester contextual features (+ all image features  $\mathbf{I}_{All}$ ). Here, one observes that the inclusion of requester contextual features other than the requester identity influences decision making quite differently between users. For instance, the correct decision rate of User *C* obtained on all requester features ( $\sim 0.8$ ) is much higher than that on only requester identity  $\mathbf{R}_I$ . For User *A* or *D*, combining different requester features ( $\mathbf{R}_{I,L,N}$  or  $\mathbf{R}_{I,N}$  respectively) generates better accuracy than just using requester identity  $\mathbf{R}_I$ . However, for User *B* and *E*, using only the requester identity  $\mathbf{R}_I$  provides the best performance, in which case the other contextual features of requester are considered as noise in machine learning. Such a variance between users again proved our hypothesis that users have different personalized behaviors in photo sharing.

## 7.4 Discussions

### 7.4.1 System Security

As is mentioned in Section 7.1.1, we assume the photo sharing service provider in ProShare S architecture is trusted. The reasons are twofold: First, it is still not possible to perform certain pattern recognition tasks on mobile devices efficiently, e.g. deep-based image semantic recognition; Second, the system makes sharing decisions in a dynamic way by analyzing both image content and requester context, which means the decision making core must lie on the service provider. However, as the development of pattern recognition on mobile devices, the security requirement of the proposed architecture can be relaxed.

In another specific case of the proposed architecture, where only requester’s identity is taken into account (no other context) in decision making, the security assumption can be discarded. In this case, the photo sharing decisions are made in a static way equivalent to predicting an access policy based on the image. According to another ProShare architecture presented in Chapter 6, such an access policy can be integrated in a CP-ABE [82] and secure photo sharing can be easily achieved through an untrusted server.

### 7.4.2 Automatic Feature Extraction

In this study, we consider mainly image semantic features, which aims to model the photo sharing decision making process of a real human: Intuitively, when one decides whether or not to share a particular photo with someone online, he/she may take into account the privacy-sensitive information in the photo, e.g. the people in the photo, activities that take place, location where the photo was captured and location information the photo may reveal. In this study, we did not tackle the automatic extraction of most image semantics; Instead, we asked subjects to manually annotate semantic information. However, thanks to recent advancements in techniques like pattern recognition, content understanding and deep learning, automatic extraction of most defined semantic features has become possible and is getting more accurate. In the following, we provide a brief review on recent research and solutions of several recognition and content understanding techniques that relate to our study.

**Face Recognition** Face recognition has been significantly developed due the recent deep learning techniques especially the convolutional neural network (CNN). Nowadays, the performance of automatic face recognition is close to human-level. Facebook presents DeepFace [91] that achieves an accuracy of 97.35% on the Labeled Faces in the Wild (LFW) dataset [92], reducing the error of the current state of the art by more than 27%, closely approaching human-level performance. DeepFace employs a nine-layer neural deep network involving more than 120 million parameters. Parkhi et al. [93] proposes another CNN-based model without any embellishments which achieves an accuracy of 98.95% slightly higher than DeepFace (both on the LFW dataset). Google presents FaceNet [94], a face recognition system that directly learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity. A new record accuracy of 99.63% has been achieved by FaceNet on the LFW dataset.

**Person Recognition** Most studies of face recognition are mainly carried out on face images. Researchers have also been working on person recognition based on not only face but also context information in image, e.g. other body parts, hair style, clothes, glasses, pose, environment and people nearby. Zhang et al. [66] proposes the Pose Invariant

PErson Recognition (PIPER) method, which accumulates the cues of poselet-level person recognizers trained by deep convolutional networks to discount for the pose variations, combined with a face recognizer and a global recognizer. [66] also creates the People In Photo Albums (PIPA) dataset, a large-scale dataset collected from Flickr photos, which consists of 37'107 photos containing 63'188 instances of 2'356 identities. Based on the test set of the PIPA dataset, an accuracy of 83.05% is achieved over 581 identities. Moreover, when a frontal face is available, PIPER improves the accuracy over DeepFace from 89.3% to 93.4%, which is close to 40% decrease in relative error. Oh et al. [95] proposes a convnet based person recognition system, which obtains so far the best results for person recognition on the PIPA dataset. This study also provides a detailed analysis of performance of different visual cues (e.g. face, head, upper body, full body, and scene) for person recognition.

**Semantic Understanding** Images contain a great amount of knowledge including the presence of various objects, the presentation of activities and scenes. Relating visual information to its semantic meanings remains an open and challenging area of research. Researchers have been working on different aspects towards the semantic understanding of image content. These studies include detection/recognition of various objects in image [96], recognition of activities from daily image [87, 97, 98], image location and scene recognition [99, 100, 101]. Particularly, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [96] was established in 2010 and has been running for six years. It has become the standard benchmark for large-scale object recognition, which consists of two components: (i) a publicly available dataset, and (ii) an annual competition and corresponding workshop. The closest to ILSVRC is the PASCAL VOC challenge [102], which provides a standardized test bed for object detection, image classification, object segmentation, person layout, and action classification. However, ILSVRC scales up PASCAL VOC's goal of standardized training and evaluation of recognition algorithms by more than an order of magnitude in number of object classes and images: PASCAL VOC 2012 has 20 object classes and 21'738 images compared to ILSVRC2012 with 1000 object classes and 1'431'167 annotated images.

**Automatic Image Tagging APIs** Automatic image tagging (also known as automatic image annotation or linguistic indexing) is the process by which a computer system automatically assigns metadata in the form of captioning or keywords to a digital image. Based on the recent developments in various image content understanding and object recognition techniques, several commercial services have been established to provide the tools or APIs for automatically assigning tags to image based on analyzing the image content. Those services includes but are not limited to: Clarifai<sup>6</sup>, Imagga<sup>7</sup>, Google Cloud

---

<sup>6</sup><https://www.clarifai.com/>

<sup>7</sup><https://imagga.com/>

Vision<sup>8</sup>, Microsoft Cognitive Services<sup>9</sup>, IBM Watson APIs<sup>10</sup>, Amazon Rekognition<sup>11</sup>, Algorithmia<sup>12</sup>, Aylien<sup>13</sup>, and Wolfram<sup>14</sup>. Those services provide different features and have their own pros and cons. Most of the services are for commercial uses but provide limited quota for free usage.

### 7.5 Conclusion

This chapter presents a conceptual architecture for context-dependent and privacy-aware photo sharing based on machine learning. The proposed architecture utilizes the images semantic and requesters contextual information to predict photo sharing decisions for users, based on their previously shared photos and past decisions. To validate the proposed model, we first conducted a user study on 23 subjects and collected a dataset containing 1'018 manually annotated images with 12'216 personalized sharing decisions in different contexts. Evaluation experiments have been performed using different classification techniques and results reveal a promising performance of the proposed architecture. Furthermore, the influence of different content- and context-related features on decision making has been investigated, which validate the importance of pre-defined features and imply a significant variance between users sharing behaviors and privacy attitudes.

Limitations of the study still remain: First, users were only put in a hypothetical photo sharing environment and given hypothesis of different sharing scenarios; Second, most image semantic features were manually annotated by subjects. Due to these facts, the semantic features we collected cannot be fine-grained. For instance, we limited the image activities in 26 keywords, which can be greatly enriched in reality. Also, we defined only a set of relationships (groups) when describing the identities in image and a prospective requester, which can be specific to individuals in practice. All these are mainly due to the lack of the access and control to a popular social network, and that automatic extraction of some semantic features (e.g. activities in image [87]) is not mature enough. However, the main aim of this study is to investigate the feasibility of using certain image semantic information and requester context in automatic photo sharing decision making. The promising results obtained provide significant sights in building accurate and reliable “privacy-aware” photo sharing decision making system based on content and context analysis.

---

<sup>8</sup><https://cloud.google.com/vision/>

<sup>9</sup><https://www.microsoft.com/cognitive-services/en-us/>

<sup>10</sup><https://www.ibm.com/watson/>

<sup>11</sup><https://aws.amazon.com/rekognition/>

<sup>12</sup><https://algorithmia.com/>

<sup>13</sup><http://aylien.com/image-tagging/>

<sup>14</sup><https://www.imageidentify.com/>

# Applications Beyond **Part III**



## 8 Towards an Animated JPEG

Animated images have recently become very popular in social networks. Alex Chung, the founder and CEO of Giphy<sup>1</sup>, one of the biggest online databases for GIF files, states that: “Eventually all Web images will be animated in Harry Potter style”<sup>2</sup>. He also says, “... the internet is sterile. It is emotionless, it lacks feeling, and its approach to content can make it feel dull.” Adopting this view point, a large fraction of still images shared over the Internet will eventually be replaced by animated images. Indeed, animation empowers users to share and express emotions in a more powerful and individualistic way. In contrast to video, animation is characterized by Moreau as: “a mini video, with no sound, that can be watched from start to finish in as little as one or two seconds in a simple, auto-looping fashion.”<sup>3</sup> Furthermore animations “offer a more convenient, faster and totally silent way to express something.” An animation is viewed in the broader context of where it appears and is therefore “the perfect combination between image and video that really captures our attention.”

Animated still image file formats such as animated Graphics Interchange Format (known as GIF) and motion JPEG provide the means to add a visually compelling motion component to still images. Compared to video files, working with still image file formats simplifies both server and client side operations and significantly reduces the computational complexity during both creation and playback. For historical reasons, animated GIF has established itself as the de facto standard for this type of content. Given the state of the art in image compression and taking into account the trend towards multimedia formats that are royalty-free, seamless and managed by international standardization committees, GIF does not seem to be the best solution to provide the necessary features in an online world demanding ever more dynamic and captivating content. Moreover, GIF supports a maximum of 256 colors (8 bit) per image frame and using GIF for photographic content reveals this limitation in the shape of severe color banding (see Figure 8.1<sup>4</sup>). On the

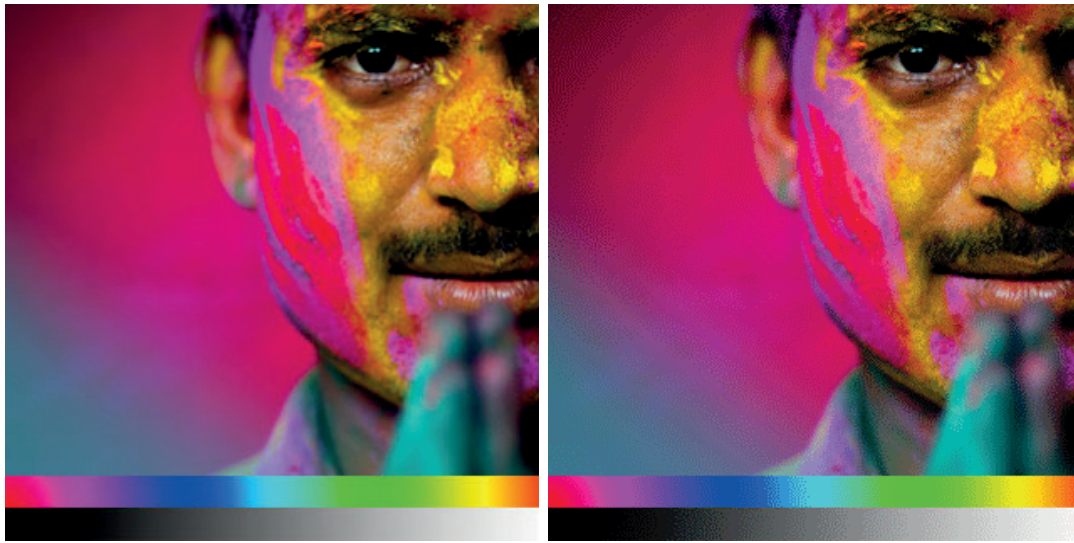
---

<sup>1</sup><http://giphy.com/>

<sup>2</sup><https://www.inverse.com/article/14908>

<sup>3</sup><https://www.lifewire.com/rise-of-animated-gif-3485813>

<sup>4</sup>Image source: <http://stackoverflow.com/questions/2336522/png-vs-gif-vs-jpeg-vs-svg-when-best-to-use>



(a) **JPEG**: 24 KB - better quality, smaller file size    (b) **GIF**: 56 KB - worse quality, larger file size

Figure 8.1 – GIF vs. JPEG.

other hand, the underlying compression used to generate GIF images is lossless and hence sub-optimal for use in consumer photographic content for the sake of storage efficiency. With animated image content now often generated on the basis of photographic images, JPEG appears to be the most suitable contender on which to base a new animated image file format.

In this chapter, we present aJPEG, an animated image format based on JPEG compression which could serve as a better alternative to animated GIF. Inspired by the JPEG Transmorphing algorithm proposed in Chapter 5, aJPEG utilizes JPEG Application Segments to preserve several frames of an animation sequence, while encoding a default frame in the main JPEG image data. Therefore, an aJPEG file is backward compatible with legacy JPEG decoders such that it allows an as seamless experience as possible to an as large as possible number of end-users. Based on the proposed aJPEG file format, we implemented an aJPEG codec and developed two prototype applications demonstrating the efficient GIF-to-aJPEG conversion and aJPEG playback.

The rest of the chapter is structured as follows. Section 8.1 presents prior work, introducing the development of GIF and other similar formats. Section 8.2 describes in detail the syntax construction and file format of aJPEG. Section 8.3 is split into two parts: the first describes an aJPEG codec, while the second describes two prototype applications of using such a format, a desktop program and a mobile application, respectively. Section 8.4 reports the performance evaluation of aJPEG with comparison to conventional animated GIF. Finally Section 8.5 concludes this chapter.



## 8.1 Prior Work

This section briefly outlines three animated image file formats, namely GIF, Motion JPEG and Motion JPEG 2000. It is not comprehensive but puts particular emphasis on their respective suitability for social media and emerging online applications.

### 8.1.1 Graphics Interchange Format (GIF)

Today, animated GIF is the dominant animated image file format. It derives its name from Graphics Interchange Format and was first introduced by CompuServe in 1987 under the name 87a. GIF was positioned as an alternative to formats such as PiCture eXchange (PCX) and *MacPaint* offering support for color and smaller file sizes. During 1989 CompuServe released version 89a of its file format which was to become known as GIF and added support for animation, transparent background colors, storage of metadata and text overlays. From the outset, GIF was subject to license and royalty constraints due to its use of Lempel-Ziv-Welch (LZW) compression [103, 104], a technology under patent by Unisys corporation until 2004 [105]. While initially released as an image file format for CompuServe customers, GIF became a de-facto standard on the early World Wide Web when support for the file format was added to the Netscape 2.0 navigator in 1995. GIF image encoding supports an input color space of 8 bits per primary color. The encoder then projects colors in the image to be encoded to a maximum of 256 colors and tabulates these in a palette which is addressed through an 8-bit index. This restricted number of colors was quite sufficient in the 1980s but severely limits the visual quality of GIF images for photographic images and content generated using modern computer graphics (see Figure 8.1). With the introduction of animated GIF, CompuServe added the concept of pixel transparency. In secondary images, pixels can be flagged to be transparent. The decoder will then substitute these transparent pixels with the corresponding pixels from the primary image. This highly desirable mechanism improves compression efficiency and in the case of computer generated content, supports more efficient and richer content creation procedures. GIF image compression is particularly suitable for sharp edges in graphical material with no noise. As such, it supports efficient encapsulation of simple graphics, logos and small, animated cartoon sequences. Today the use of GIF extends to sprites and avatars in online games and mobile apps. With social network platforms such as Facebook, adding support for GIF, users are widely adopting the opportunity to post animated images, giving GIF a new lease on life. The bitstream syntax used in GIF file format is shown in Figure 8.2 and a description of GIF format can be found on this website [106]. The full specifications for the GIF file format is available at <https://www.w3.org/Graphics/GIF/spec-gif89a.txt>.

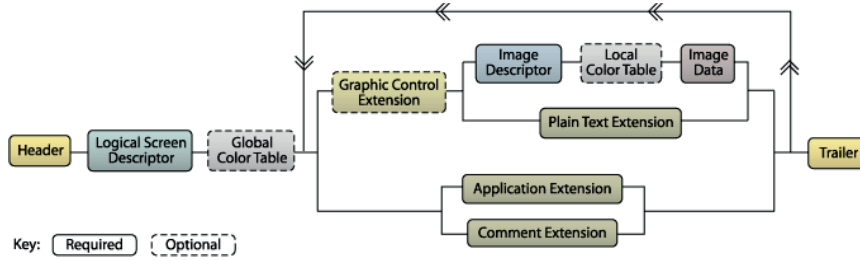


Figure 8.2 – Syntax of GIF file format.

### 8.1.2 Other Animated Image Formats

Motion JPEG (MJPEG) addresses the sub-optimal compression in GIF by replacing LZW with the efficient image compression embodied in the JPEG standard [59]. The basic architecture of an MJPEG file follows that of a GIF file: Each image constituting an MJPEG sequence is individually compressed using JPEG. These images are then combined into a single file which additionally includes display, context and metadata. As such, MJPEG is an intraframe coding scheme and no temporal redundancy in the image sequence is exploited to achieve higher compression. The MJPEG file for a given image sequence is therefore larger than the equivalent sequence obtained from state-of-the-art video compression scheme such as H.265/HEVC. But this is offset by much reduced complexity, both during encoding and decoding. Many variants and implementations for MJPEG have been put forward. Yet, no unifying standard or specification was ever adopted and published by the JPEG standardization committee for an MJPEG file format. As a consequence, MJPEG has not achieved widespread acceptance due to incompatibility and interoperability issues which exist between implementations put forth by different developers and system vendors.

Motion JPEG 2000 follows the same implementation strategy as animated GIF and MJPEG, building on a simple file structure and an intraframe coding strategy for each image in the sequence. Avoiding the fragmentation that blocked the widespread adoption of MJPEG, the JPEG 2000 standard was adopted and published by JPEG standardization committee in a Part 3 extension of JPEG 2000 standard specifying support for motion and animation<sup>5</sup>. Yet JPEG 2000 and, as a consequence, Motion JPEG 2000 are victims of the success of the legacy JPEG standard which, today, is still the dominant image coding standard employed in all consumer and prosumer products. This underlines the requirement for backward compatibility to JPEG of a motion image format that could serve to displace GIF in Internet and social media applications.

Furthermore, other file formats such as Animated Portable Network Graphics (APNG) [107] or Multiple-image Network Graphics (MNG) [108] have been put forward. But none has attained GIF's popularity and widespread use despite two major shortcomings of GIF.

<sup>5</sup><https://jpeg.org/jpeg2000/>

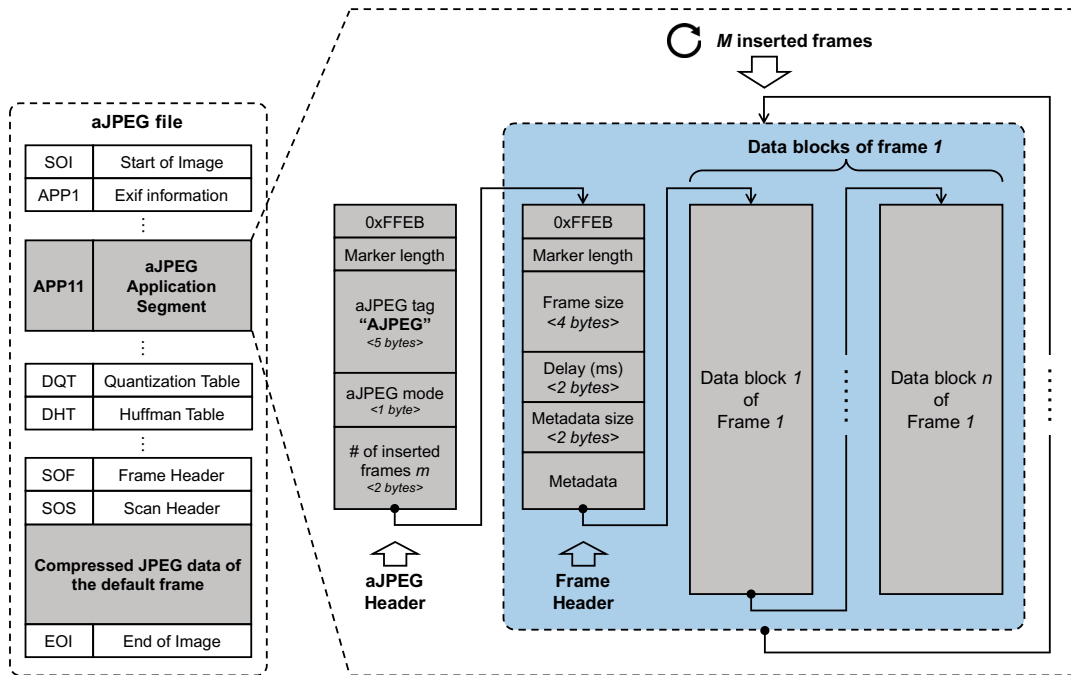


Figure 8.3 – Syntax of aJPEG file format

## 8.2 aJPEG Syntax and Structure

### 8.2.1 aJPEG Overview

The central principle behind the proposed animated JPEG is to encode a default frame of an animation sequence as standard JPEG while storing the information about the other frames in the Application Segments of the JPEG file header. Therefore, any legacy JPEG decoder will be able to decode the default frame. To decode the other frames and display the animation, a dedicated aJPEG decoder or transcoder is required. The syntax of an aJPEG file is shown in Figure 8.3. An aJPEG image file starts with a standard JPEG header, consisting of various APP markers for different purposes. We then took a unique APP marker for the use of aJPEG. In our current implementation, APP11 is used. The metadata and image data of the animated image frames, except for the default frame, is inserted in a sequence of APP11 markers. The inserted data consists of the necessary information to recover the animated image frames, including the number of inserted frames, frame rate, and the compressed image data of those frames. Finally, the image data of the default frame is compressed in JPEG in the main aJPEG file, which can be read by any legacy JPEG decoder. In our current implementation, the inserted animation frames are compressed with the same JPEG quality factor as the default frame. Details of the format syntax and structure are described in the following.

### 8.2.2 aJPEG Header

The metadata of the inserted animation frames starts as an aJPEG header, the structure of which is shown in Figure 8.3. The aJPEG header uses an APP11 segment to signal the basic information about the inserted frames. As any JPEG marker, it starts with an marker ID `0xFFE8` and the marker length, a header tag “AJPEG” identifying the aJPEG format. This tag occupies five bytes in the segment. Then, the next byte is used to signal the mode of an aJPEG format, which indicates the method for encoding the inserted image frames. In our current implementation, for instance, all inserted frames are compressed in JPEG with the same quality factor as is used for the default frame. In practice, any other encoding methods can be used to compress the inserted frames. At the end of the aJPEG header, two more bytes are used to indicate the number of inserted image frames, noted as  $M$ .

### 8.2.3 aJPEG Inserted Frames

After the aJPEG Header, a sequence of segments are used to store the image data of animated frames one by one. Each image frame starts with a frame header, followed by a sequence of segments storing the image data of that frame.

**Frame Header** The structure of a frame header is illustrated in Figure 8.3. Similar to any other APP marker, this header starts with the marker ID and marker length. The rest of this header signals the following metadata of the particular frame:

- **Image size  $s$ :** file size of the frame in bytes.
- **Frame delay:** the duration (in hundredths of a second) for which the frame is displayed during animation.
- **Extra metadata:** extra information about the frame, e.g. annotation and geo-tag.

**Frame Data Blocks** After the frame header, the bitstream of each compressed image frame is inserted in a sequence of segments byte by byte. Since JPEG allows APP segment no longer than 65,535 bytes, as explained in Chapter 5, the bitstream of each compressed image frame might need to be separately stored in  $n$  APP markers:

$$n = \lceil \frac{s}{65533} \rceil, \tag{8.1}$$

where  $s$  is the size of the frame in bytes and  $\lceil \cdot \rceil$  is the ceiling function.

### 8.2.4 aJPEG Compressed Image Data

The image data of the default frame is compressed in standard JPEG, consisting of color space transformation, downsampling, discrete cosine transform (DCT), quantization, and entropy coding. Since the aJPEG file still contains the standard JPEG headers, a legacy JPEG decoder can read the compressed default frame as a standard JPEG file. With a dedicated aJPEG decoder or transcoder, the other image frames can be read from the aJPEG APP markers and played back as an animated sequence. In other words, an aJPEG file is backward compatible with legacy JPEG standard, with the advanced functionality of being rendered as animated content.

## 8.3 Codec and Prototype Applications

Based on the proposed aJPEG file format, we implemented a dedicated aJPEG codec for encoding and decoding of aJPEG images, based on an open source JPEG library version 6b maintained by the Independent JPEG Group (IJG)<sup>6</sup>. To demonstrate the use of aJPEG, we developed two prototype applications for GIF-to-aJPEG conversion and aJPEG playback, based on personal computer and smart phone, respectively.

### 8.3.1 aJPEG Codec

The aJPEG codec consists of two basic components, i.e. an encoder and a decoder. The encoder takes as input a sequence of JPEG images and converts them to an aJPEG file with specific format defined in Section 8.2. On the contrary, the decoder extracts all frames from an aJPEG file. The aJPEG codec is implemented as a command line executable named `ajpegtran`, which has two basic modes `build` and `extract` for encoding and decoding respectively.

### 8.3.2 Prototype Applications

To illustrate the use of the proposed aJPEG format, we developed two prototype programs that can (i) convert a GIF to an aJPEG file and (ii) play back animations given as input an aJPEG file. The two programs are both implemented in Java based on desktop (Mac OS X) and mobile phone (Android) respectively. In both applications, one can load an input file, in either GIF or aJPEG format. If an animated GIF image is loaded (GIF-to-aJPEG conversion mode), the program converts the GIF to a sequence of image frames in JPEG and further converts all those frames to an aJPEG format file. Once finishing the conversion, it plays back the animations. While GIF-to-aJPEG conversion, the quality factor can be set by user with 75 by default. The frame delay of generated aJPEG is the same as the original GIF. Furthermore, in the GIF-to-aJPEG conversion

---

<sup>6</sup><http://www.ijg.org/>

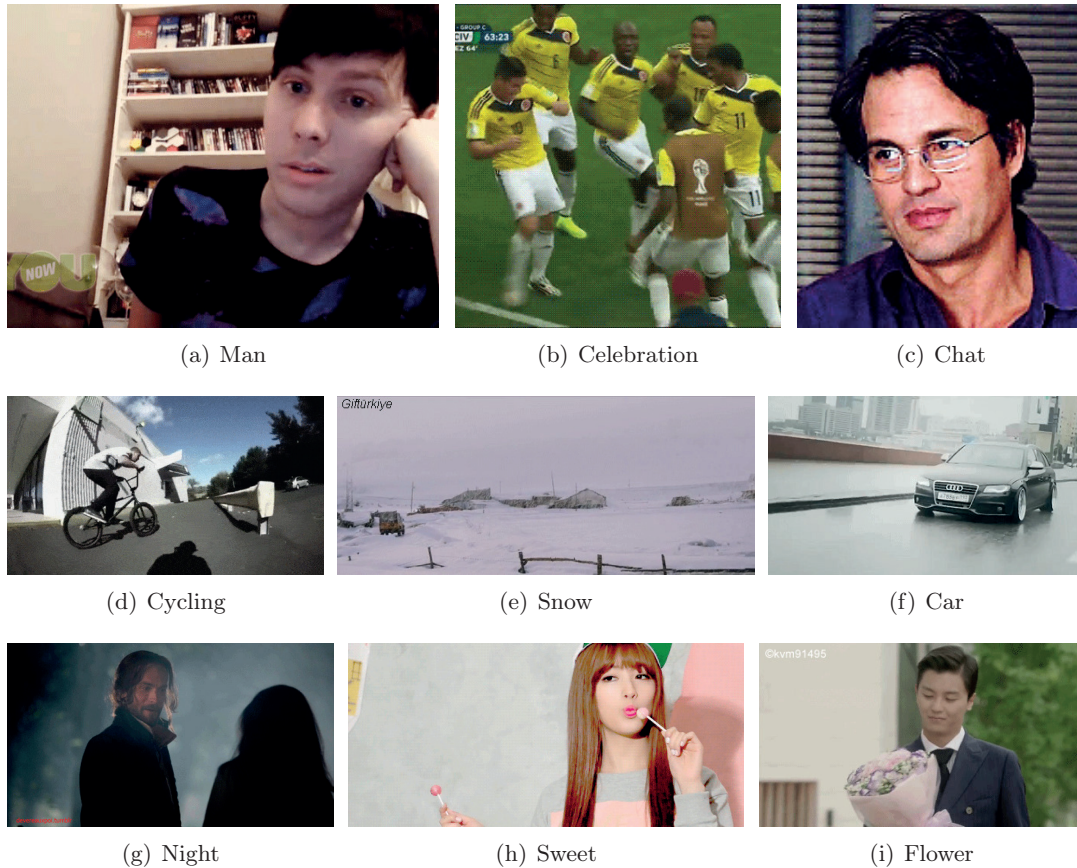


Figure 8.4 – Nine GIF images selected from TGIF dataset.

mode, the program displays the file sizes and ratio of input GIF and output aJPEG file. If an aJPEG image is loaded, the program (in playback mode) simply decodes each frame and plays back the animations. In this case, the program acts as a simple aJPEG viewer or player. In playback mode, user can set a speed factor to control the playback speed (frame rate). Screenshots of the two applications are shown in Figure A.7 in Appendix A.

## 8.4 Performance Evaluation

This section reports performance evaluations of the proposed aJPEG format in comparison to GIF, with respect to compression ratio and image quality.

### 8.4.1 Datasets

Two datasets were used in the experiments:



Figure 8.5 – Six video sequences from the EPFL-PoliMI dataset.

- **Tumblr GIF Description Dataset (TGIF)**<sup>7</sup> [109]: This dataset contains 100K animated GIFs and 120K sentences describing their visual content. We selected nine GIF files from the whole dataset, which cover different types of image content. The nine GIF images are named *Man*, *Celebration*, *Chat*, *Cycling*, *Snow*, *Car*, *Night*, *Sweet* and *Flower* respectively. Example frames of the nine animations are shown in Figure 8.4.
- **EPFL-PoliMI Video Quality Assessment Database**<sup>8</sup> [110, 111]: This dataset contains 156 video streams for video quality assessment. From the dataset, we took six video sequences which were also used in subjective evaluations in [111]. The six sequences are all in I420 raw progressive format, with 10 seconds long at 4CIF spatial resolution (704×576 pixels). They are referred to as *Crowdrun*, *Duckstakeoff*, *Harbour*, *Ice*, *Parkjoy* and *Soccer* respectively. Example images of the six video sequences are shown in Figure 8.5.

### 8.4.2 Compression Ratio

In the first experiment, we evaluate the compression ratio of aJPEG images compared to GIFs, in terms of their file sizes. First, we converted each of the nine GIFs from TGIF

<sup>7</sup><http://raingo.github.io/TGIF-Release/>

<sup>8</sup><http://vqa.como.polimi.it/>

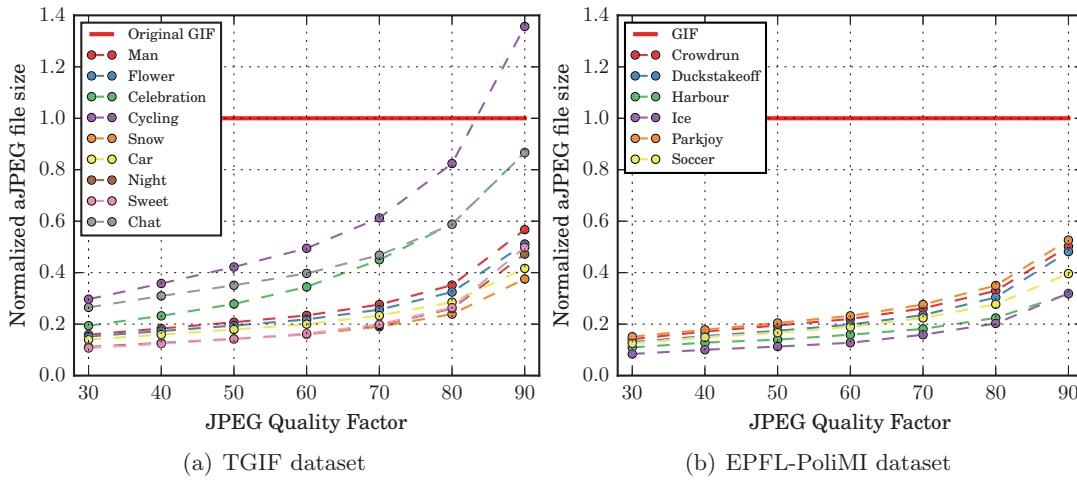


Figure 8.6 – Normalized aJPEG file size compared to GIF.

dataset into a set of aJPEG files compressed with different quality factors  $Q \in [30, 90]$ . The conversion is done by using the GIF-to-aJPEG converter in Section 8.3. For each aJPEG file, we computed its normalized file size compared to file size of the original GIF image, i.e. file size of aJPEG/file size of GIF. The normalized file sizes of the nine images are shown in Figure 8.6(a). From the results, one observes that the file sizes of most aJPEG images (except for *Cycling*) compressed with  $Q = 90$  are smaller than the original GIF. With smaller quality factor applied, the file sizes of aJPEG files are further reduced and all become smaller than GIF. In  $Q = 80$ , the normalized sizes of six images are between 0.2 and 0.4 with three images (*Cycling*, *Chat* and *Celebration*) as exceptions. This means that the compression ratio from GIF to aJPEG also depends on image content.

A similar experiment was conducted using the six video sequences from EPFL-PoliMI dataset. This time, for each sequence, we converted the first 100 raw image frames to an animated GIF file and different aJPEG files compressed with varying quality factors ( $Q \in [30, 90]$ ), with the same frame rate. The normalized file sizes of aJPEG files are shown in Figure 8.6(b). Similar observation is found: For every video sequence, the file size of its aJPEG version is always significantly smaller than that of GIF, even at the JPEG quality of 90. When applying a quality factor of 75, the compression ratio is up to 0.2. Considering that JPEG compression with a quality factor of 75 usually provides good-enough visual quality, aJPEG outperforms GIF in terms of compression ratio.

### 8.4.3 Image Quality

To further evaluate the performance of aJPEG with respect to its image quality, we conducted another set of experiments to compare aJPEG and GIF, using two objective



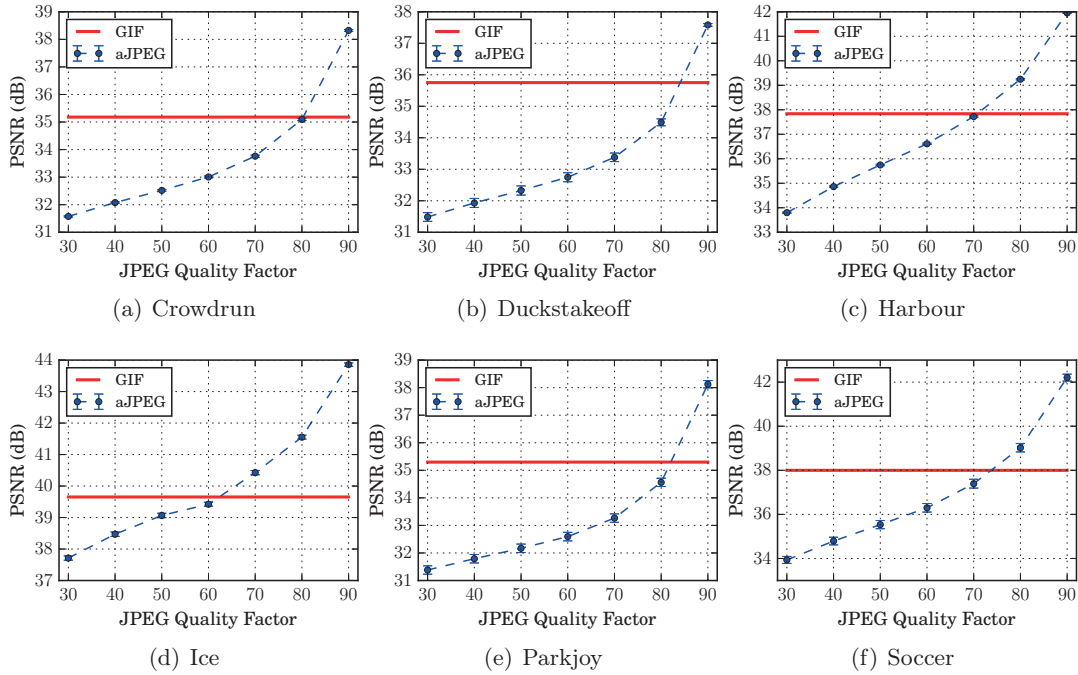


Figure 8.7 – Comparison of PSNR between aJPEG and GIF.

metrics: peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [73]. In this experiment, we used the six raw video sequences from EPFL-PoliMI dataset, and generated from raw video the GIF and aJPEG files in the same way as is in Section 8.4.2. Then, we computed the PSNR and SSIM of each GIF and aJPEG frame compared to the original raw video frame. For each content, only the first 100 frames are considered. The PSNR and SSIM for GIF (mean) and different aJPEG files (mean and 95% confidence interval) are show in Figure 8.7 and Figure 8.8 respectively.

For every image content, aJPEG frames compressed with the quality factor of 90 always show better quality than GIF in both metrics. For most video sequences, aJPEG coded with JPEG Q factor of 80 reveal better or similar quality compared to GIF in terms of both PSNR and SSIM. According to the results from Section 8.4.2, aJPEG file compressed at Q factor of 80 or 90 has already much smaller file size than GIF; in this regard, we consider Q factor between 80 and 90 as a promising quality factor to reach a good balance between file size and image quality of aJPEG. Combining the results of both experiments on compression ratio and image quality, aJPEG shows a significant advantage compared to animated GIF: With a proper quality factor selected, aJPEG could serve better quality but much smaller file size.

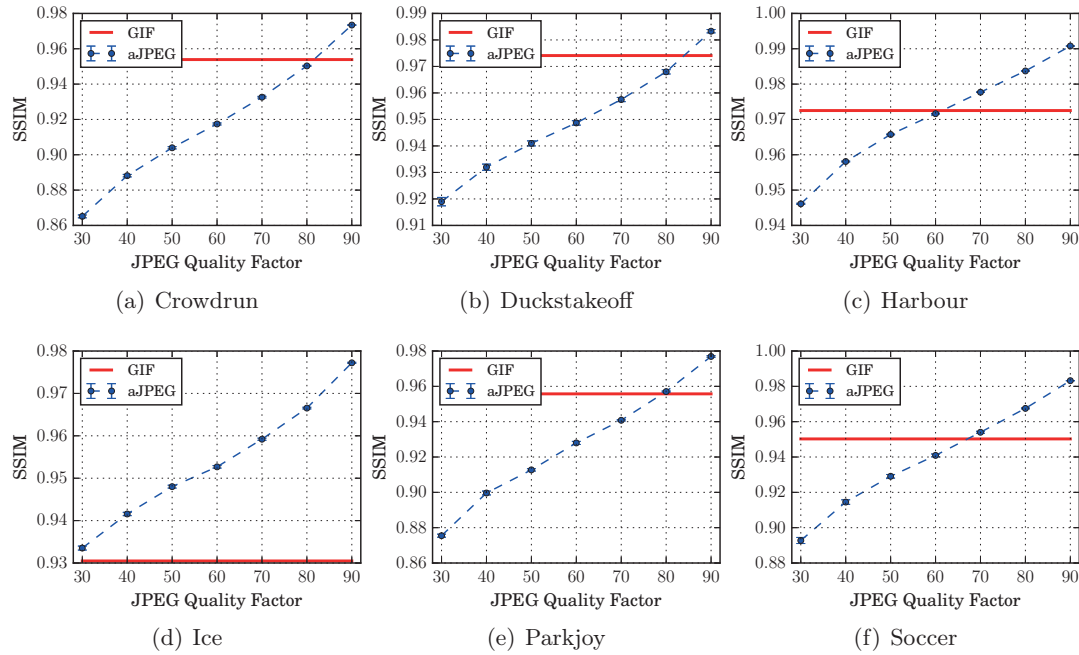


Figure 8.8 – Comparison of SSIM between aJPEG and GIF.

## 8.5 Conclusion

This chapter presents aJPEG, a novel animated image format based on JPEG compression. Adopting the similar idea of JPEG Transmorphism presented in Chapter 5, aJPEG encodes a default frame selected from an animation sequence in a standard JPEG image while preserving the data of other frames in JPEG Application Segments of the “cover” image. Therefore, an aJPEG file is backward compatible with JPEG. Any legacy JPEG decoder or viewer is able to decode and display the default frame of an aJPEG file. Only with a dedicated aJPEG decoder, the other frames can be extracted and displayed as an animation. Based on the proposed aJPEG file format, we implemented an aJPEG codec and developed two prototype applications demonstrating the GIF-to-aJPEG conversion and aJPEG animation playback. Moreover, we conducted two experiments to evaluate the performance of aJPEG in regard to its compression ratio and image quality. The experiments were performed in comparison with GIF, and results show that aJPEG image compressed with JPEG quality factor of higher than 80 (preferable 90) outperforms conventional GIF in terms of both file size and image quality. Considering the wide popularity of JPEG, such an encoding method for animated image could serve as a better alternative to conventional GIF, especially for presenting photographic image content in the scenario of Internet and social media.

## 9 Understanding Emotional Impact of Image Manipulation

Modern photo sharing applications are usually equipped with interesting, easy-to-use and even interactive image editing tools. Famous applications include Instagram, Snapchat and so on. Those applications provide consumers with convenient solutions to make their pictures more attractive, and more importantly, to arouse stronger emotional resonances. Different types of image content generate different emotions. Using different photographic techniques, visual filters or editing tools, pictures of the same scene can also evoke different emotions. Motivated by these facts, we attempt to change an original picture's evoked emotion and transform it to new emotions (stronger, weaker, or completely different) by image manipulation. To achieve this goal, we first need to understand the emotional responses evoked by different image manipulations when applied to pictures.

This chapter investigates the influence of image manipulations on evoked emotions, and aims to find the potential pattern between image content, manipulation and generated emotions. To do so, we conducted subjective experiments based on online crowdsourcing. Different types of images were collected from Instagram, and manipulated by a number of typical image editing tools. Subjects were then exposed to each, and questioned regarding their emotions pictures induced on them. Using the crowdsourced data as groundtruth, we train and evaluate a simple regressor for predicting evoked emotions, taking as input an original image and the desired manipulation.

The rest of the chapter is structured as follows. Section 9.1 outlines prior work in the field of image emotion analysis, classification and recognition. Section 9.2 describes our user study and collected data. Section 9.3 and Section 9.4 analyzes emotional responses obtained from user study and reports the experiments on emotion prediction upon image manipulation. Finally, Section 9.5 concludes this chapter and discusses future work.

### 9.1 Prior Work

Image aesthetic quality estimation, emotion recognition and classification have been largely studied in the field of computer vision [112, 113, 114, 115, 116]. Most previous works use image features for affective image classification and emotion prediction [113, 114, 117, 118, 116]. Such features include color, texture, composition, edge and semantic information. A few researchers have worked on transforming image emotions by editing images. In [119], Wang et al. associate color themes with emotion keywords depending on art theory and transform the color theme of an input image to the desired one. However, in their work, only a few cartoon-like images are used. Peng et al. [120] propose a framework to change an image’s emotion by randomly sampling from a set of possible target images, but only show a few examples. Jun et al. [121] show that changing brightness and contrast of an image can affect the pleasure and excitement felt by observers. However, only a limited variation of an input image can be produced by changing the two features. Peng et al. [122] change the color tone and texture related features of an image to transfer the evoked emotion distribution, with experiments conducted on only limited types of image content.

Evaluating image’s evoked emotions after image manipulation is not a trivial task. Many well-established image manipulation and editing tools have been widely used in online photo sharing and social networks, as ways for users to enhance their image content either to draw better attention or to evoke stronger emotions. Popular image editing tools include image enhancement [123], grayscale conversion, vintage processing, cartoonizing [124], and more recently addition of stickers<sup>1</sup> [125]. However, most image manipulation methods have been studied merely from the perspective of image processing and not so much on their emotional impact.

In the previous studies on affective image classification or emotional response prediction, various types of images have been experimented with. Machajdik et al. [113] used artistic photos or abstract paintings for affective image classification. Peng et al. [122] collected an image dataset named Emotion6 containing only images without high-level semantic features, such as human facial expressions or text. In addition, the other older affective image datasets like International Affective Picture System (IAPS) [126] and Geneva Affective Picture Database (GAPED) [127] both have only limited content types. In our research, we are more interested in the emotions of everyday photographs, especially those images that are widely shared by online users. Unfortunately, most existing affective image datasets contain either extremely emotional images with special content or images without much natural high-level semantic information. All those types of images do not fit our requirements. Therefore we decided to collect our own dataset using Instagram, one of the most popular online photo sharing services.

To measure emotions, different types of models have been designed by psychologists.

---

<sup>1</sup><https://www.facebook.com/help/1597631423793468>



Figure 9.1 – An example image processed by seven different manipulations.

One of the most popular is the valence-arousal (VA) model (proposed by Russell [128]), characterizing emotions in two dimensions, where valence measures attractiveness in a scale from positive to negative, while arousal indicates the degree of excitement or stimulation. In terms of categorization of emotions, Ekman’s six basic emotions (anger, disgust, fear, joy, sadness and surprise) [129] are widely known. In this study, we used both models similar to the studies in [127, 122].

## 9.2 Image Dataset and User Study

### 9.2.1 Image Collection and Preprocessing

We collected images from Instagram, one of the most popular photo sharing platforms. According to a previous study by Hu et al. [86], images shared within Instagram can be classified into the following eight basic categories in terms of their content: *Friends*, *Food*, *Gadget*, *Captioned photo*, *Pet*, *Activity*, *Selfie* and *Fashion*. Therefore, we utilized this categorization and we collected the image dataset by searching for the eight category keywords or their synonyms via Instagram #tag. This was mainly motivated in order to have a wider variety of image content. Unlike the work by Peng et al. [122], we are

concerned with all kinds of images people daily capture and share including those ones with high-level semantic features. These features may have significant influence on evoked emotions, but such influence along with applied manipulations has not yet been well understood. At the end, 13 color images were selected for each image category resulting in 104 images in total. All selected images have the same size of  $640 \times 640$  pixels. For each image, we applied seven manipulations to create different visual effects. We refer to the seven manipulations as follows in the rest of the chapter:

- **Cartoon:** Applies a cartoonized effect to an image.
- **Emoji:** Adds an “Tear of Joy” Emoji sticker on top-right corner of an image.
- **Enhance:** Applies brightness/contrast/colorization enhancement on an image via LAB colorspace.
- **Halo:** Applies a circular halo effect to an image.
- **Gray:** Converts an image to gray scale.
- **Grunge:** Applies a classic vintage effect with a grunge background to an image.
- **Old paper:** Applies another heritage vintage effect with an old paper as background.

The reason of selecting the particular seven manipulations is that they modify image visual information from very different aspects, including color, texture, composition and higher-level image semantics. The emoji sticker “Tear of Joy” was selected as it has been in the top 10 most popular emojis on Emojipedia<sup>2</sup> for all of 2015, and the emotion it expresses is not very straightforward. The seven manipulations were implemented using the ImageMagick software<sup>3</sup>. An example image processed by the 7 different manipulations is illustrated in Figure 9.1. Summing up, a grand total of 832 ( $104 \times 8$ ) images were generated, including the original versions of each image. The image dataset is publicly accessible at <http://mmspg.epfl.ch/emotion-image-datasets>.

### 9.2.2 User Study based on Crowdsourcing

We used Microworkers<sup>4</sup> platform to collect emotional responses from subjects. A questionnaire was designed where four emotion-related questions are asked for each image. The first two questions are about the valence and arousal ratings respectively, where a 9-point scale was used, same as [122, 126]. For valence, 1, 5, and 9 mean very negative, neutral, and very positive emotions respectively, in terms of attractiveness. For arousal, 1 and 9 mean emotions with very low and very high stimulating effects respectively. In

---

<sup>2</sup><http://emojipedia.org/face-with-tears-of-joy/>

<sup>3</sup><http://www.imagemagick.org/script/composite.php>

<sup>4</sup><https://microworkers.com/>

the questionnaire, instead of directly asking subjects to provide VA scores, questions are rephrased to be similar as in [122]. The third question is about the emotion distribution of the image, based on Ekman’s six basic emotions [129]. Similar to [122], 7 emotion keywords (Ekman’s six basic emotions and “Neutral”) are used and subjects are asked to select the keywords that best describe their emotions after seeing a particular picture. In the fourth question, we ask subjects to select the content-related factors that have the most influence on their emotional decisions. The 7 pre-defined factors are:

- **Face:** Human facial expression, post, gesture, etc.
- **Color:** Image color, contrast, saturation, etc.
- **Scene:** Image background, scene, or any landmark.
- **Object:** Objects in image, such as gadget, clothes and also animals.
- **Text:** Texts in image.
- **Emoji:** The Emoji sticker in the image.
- **Halo:** Halo effect applied to the image.

We gather these information in order to further understand how the image content and manipulation jointly influence evoked emotions. Furthermore, in every questionnaire, a number of CAPTCHA questions (e.g., “ $56 + 78 = ?$ ” and “If the arm is green, what color is it?”) were included to detect and remove subjects who provided sloppy answers. Each questionnaire contains eight images, selected in a semi-random way such that the following constraints are satisfied: (i) The eight images in each questionnaire came from eight different categories respectively; (ii) The order of image manipulations appearing in each questionnaire was randomized. Therefore it ensured that subjects saw different image content in each questionnaire with manipulations in different orders. We aimed at collecting twenty answers for each questionnaire, meaning that every image was to be rated by 20 different answers. Therefore, a total of 2080 ( $20 \times 832/8$ ) questionnaires (implemented using internal template by Microworkers) were distributed online. Thanks to Microworkers’ rating system, we kept tracking the results and ruling out answers from dishonest subjects while task campaigns were running, based on their answers to CAPTCHA questions and the time spent on each questionnaire (for those who spent less than 120 seconds to answer a questionnaire, their answers were removed). The vacated positions were then taken by new subjects until their answers satisfied the above requirement. Finally, answers from 590 unique subjects were collected, each rating 28.2 images in average. Screenshot of an image under evaluation in a questionnaire is shown in Figure A.8 of Appendix A.

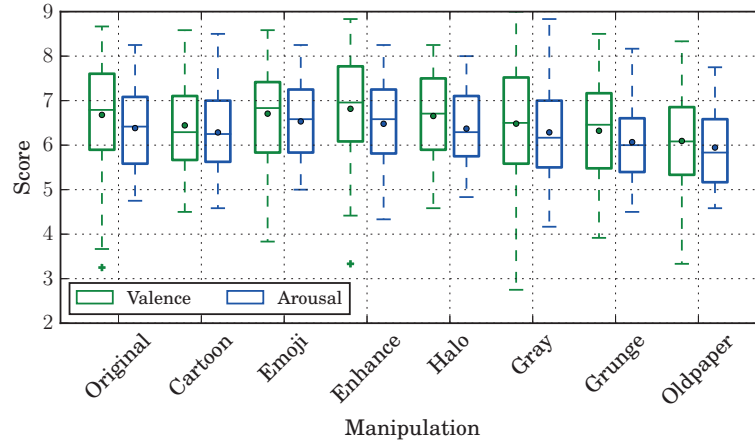


Figure 9.2 – Boxplot of overall VA scores for each image manipulation method.

### 9.3 Analyzing Emotions induced by Image Manipulation

In this section, we analyze the emotional responses obtained from the crowdsourcing user study, with respect to each question.

#### 9.3.1 Valence-Arousal Score

Firstly, for each image, the mean valence and arousal scores are computed, by averaging all the rated VA scores. Then, for each image manipulation (including the original), distributions of all images mean VA scores are gathered and plotted with box plot in Figure 9.2. Among all the manipulations, vintage processing with “Grunge” and “Old paper” generate the lowest VA scores in general. Besides, for certain methods such as “Gray”, VA scores show a higher variance than that of other methods. The other manipulations influence the evoked VA scores in different degrees. However, this is just a first glance at the overall VA distributions of all images. We take as working hypothesis that manipulating an image in a certain way leads evoked emotions to change along a certain direction, but the change of emotions due to image manipulation highly depends on image content.

To verify this assumption, we investigate the influence of the two factors (image content and manipulation) on evoked emotion, with respect to VA scores. We compute the mean VA scores of each image category for different manipulation methods, by averaging the scores of 13 pictures belonging to each category for each method. Then we present the difference VA scores ( $\Delta VA = \text{original score} - \text{score after manipulation}$ ), with respect to different image categories, as heat maps shown in Figure 9.3. The higher the absolute value of the  $\Delta VA$  score, the more influence the manipulation has on evoked emotion. From the results, one observes that certain image manipulations have greater impact on evoked emotions of certain types of images. For example, manipulations “Grunge” and



### 9.3. Analyzing Emotions induced by Image Manipulation

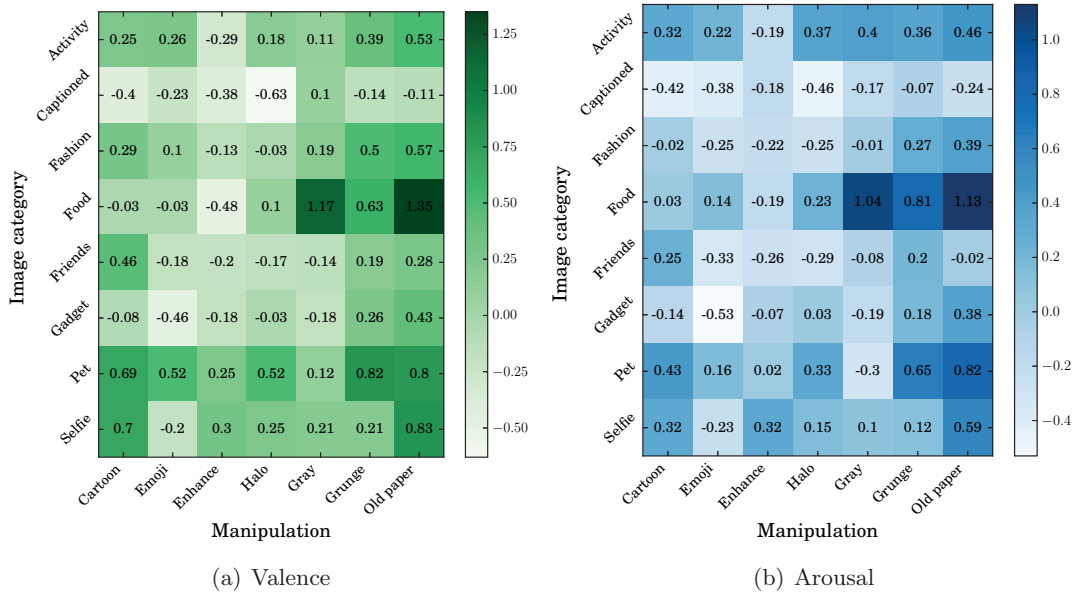


Figure 9.3 –  $\Delta$ VA scores for different image content and manipulations.

“Old paper” greatly lower the VA scores of most image content, especially for “Food” and “Pet” images. In addition, other manipulations like “Emoji” and “Halo” both increase the VA scores of “Gadget” and “Captioned” images. We then plot the  $\Delta$ VA scores of every image (difference score between original and manipulated version by “Old paper”), versus the their original VA scores, as a scatter plot shown in Figure 9.4. Here, we observe that images with higher original valence or arousal scores are more likely to generate higher difference VA scores, indicating that images with higher VA scores are prone to be impacted by the manipulation.

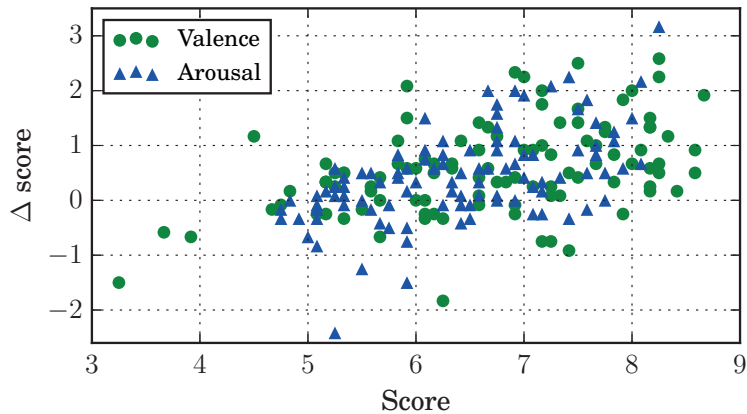


Figure 9.4 – Scatter plot of all  $\Delta$ VA scores due to “Old paper” manipulation.

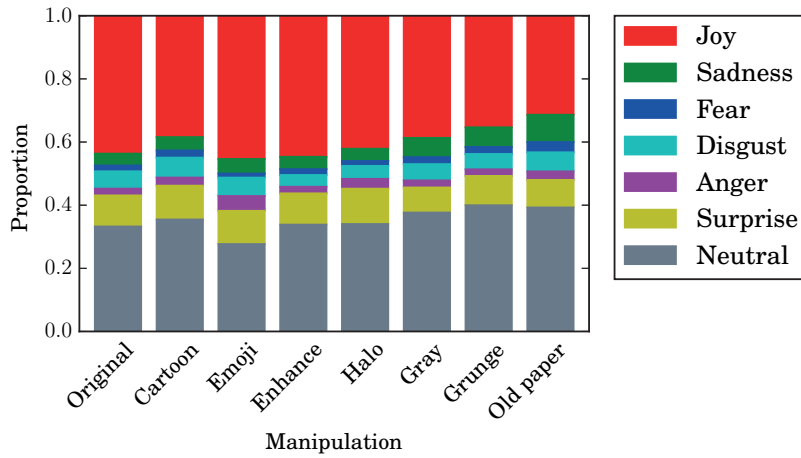


Figure 9.5 – Average emotion distribution of different manipulation methods.

### 9.3.2 Emotion Keywords Distribution

We then assess the evoked emotions in terms of probability distribution of emotional keywords. To obtain the emotion distribution of each image, we counted the occurrence of each emotion keyword voted by subjects on each image, and generate a normalized distribution over the 7 keywords, by dividing the number of keyword with the total number of voted keywords. The average emotion distribution corresponding to each manipulation method is shown in Figure 9.5. Again, one observes overall changes on the emotion distributions across different manipulations. The most significant observation is that the proportion of emotion “Joy” for images processed by the two vintage filters (“Grunge” and “Old paper”) is much reduced compared to the original images. At the same time, subjects have been evoked more “Sadness” and “Neutral” emotions by the two methods. In particular, “Emoji” causes drastic changes on emotion distributions, where the proportion of positive emotion “Joy” and negative emotion “Anger” are both increased.

To quantify the changes of emotion distributions due to different manipulations, we used two metrics Euclidean Distance (ED) and Chebyshev Distance (CD) to compute the difference in emotion distributions between original and manipulated images. The average distances for different image categories and manipulations are plotted as heat maps in Figure 9.6. This time, one observes again that different types of image content are influenced by image manipulation methods in different degrees. Similar to the previous results on VA scores, methods “Gray” and “Old paper” generate higher distances in emotion distributions for many types of images, especially “Food” and “Pet”. Among all the image content, “Fashion”, “Food”, “Gadget” and “Pet” are prone to be impacted by manipulations than other types of content.

## 9.4. Predicting Emotions induced by Image Manipulation

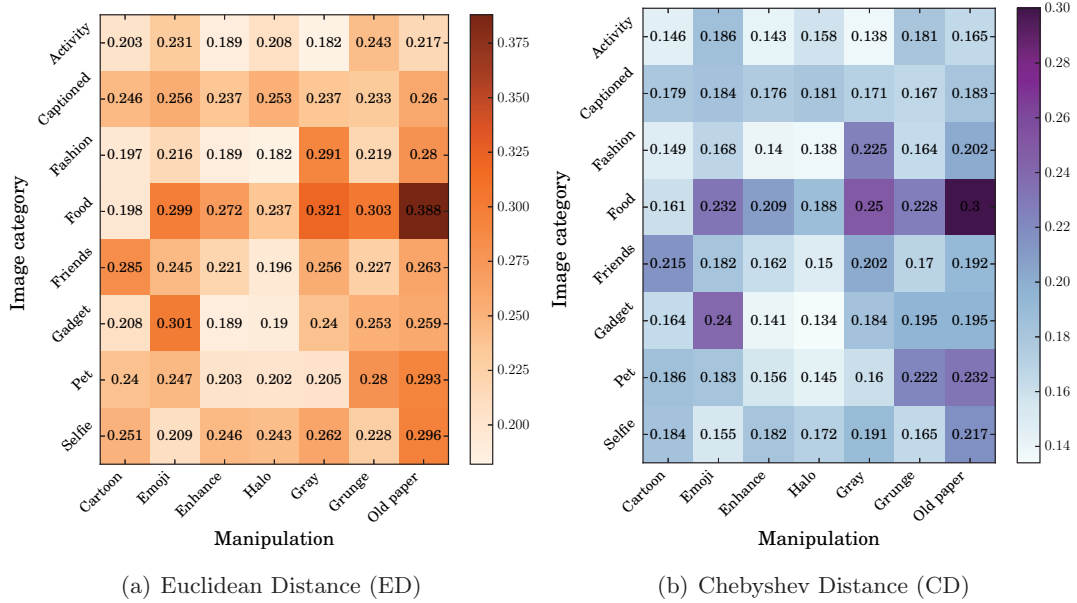


Figure 9.6 – Difference in emotion distributions between original and manipulated images.

### 9.3.3 Influence of Image Content

Furthermore, we investigate the influence of image content on evoked emotions. This is to understand how different types of image visual cues and image manipulation jointly influence evoked emotions. To do so, we compute the overall number of each content-related factors selected by subjects. The number of each factor for each manipulation (including original image) is plotted in Figure 9.7. In all manipulations, factors like “Face”, “Color” and “Object” highly influence subjects’ evoked emotions, followed by the “Scene” and “Text”. However, certain image manipulations that modify high-level image semantic information can draw subjects attentions and facilitate their decision making as well. Such example are those images manipulated by “Emoji” and “Halo”, where the influence of Emoji sticker and halo effect are significantly improved. In addition, one observes that the influence of “Color” varies significantly between different manipulations. For instance, the “Color” information has clearly more impact on any manipulation except for “Emoji” and the original image. This is because those manipulations change image color information drastically compared to the original image and the “Emoji” masked image.

## 9.4 Predicting Emotions induced by Image Manipulation

As stated in the beginning of the chapter, the final aim of this study is to manipulate an original picture and transform its evoked emotions to new emotions (stronger, weaker, or completely different). To this end, it requires a sort of mechanisms that are able

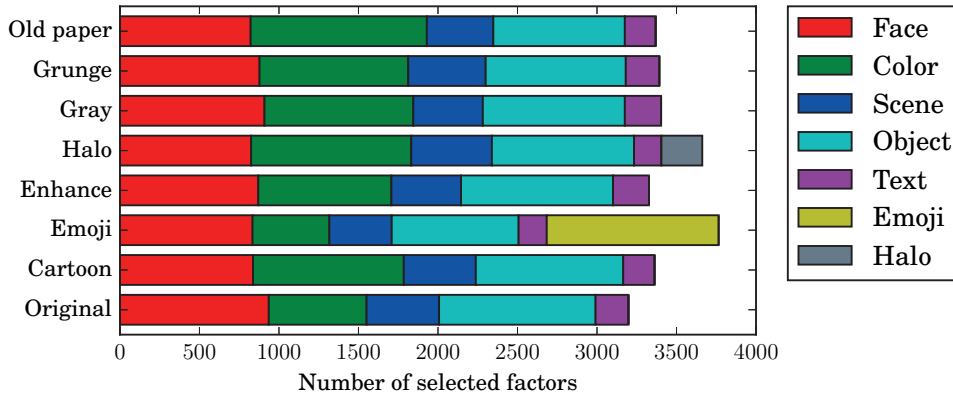


Figure 9.7 – Number of influential factors for different manipulations.

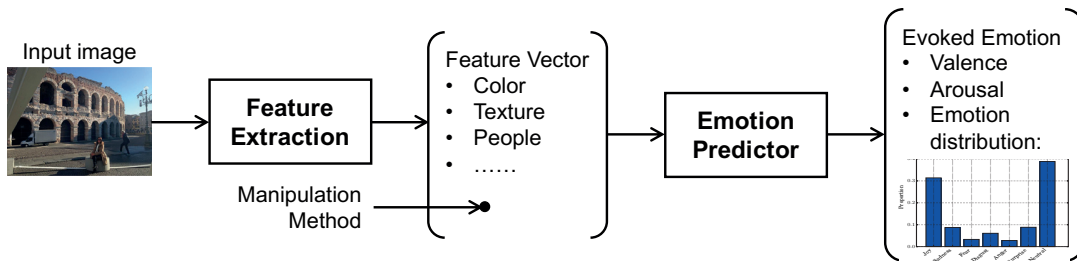


Figure 9.8 – Framework of an emotion prediction system.

accurately predict the emotion of an image manipulated by certain editing, even before the manipulation has been applied on the original image. The advantages are twofold: First, it can predict the emotions of an image after being manipulated by a desired editing method without actually applying the manipulation. This is especially good for computational costly operations, such as the recently popular image editing app Prisma<sup>5</sup> that applies artistic style image morphing [130]. Second, the proposed predictor extracts image features only once from the original image, which does not require the system to extract features each time from a newly manipulated image. Both advantages could help use make an advanced image emotion transformation system that can recommend the image manipulation method based on image content and the desired emotion.

Therefore, in this section, we conduct experiment on a predictor that can estimate the emotions of the manipulated version of an image given only the original image and the desired manipulation method as input. The purpose of this experiment is to investigate the feasibility of accurately predicting image emotions according to the prediction mechanism as is stated and find the potential pattern between image content, manipulation and evoked emotions.

<sup>5</sup><http://prisma-ai.com/>

<sup>6</sup><https://ch.mathworks.com/help/images/gray-level-co-occurrence-matrix-g lcm.html>

## 9.4. Predicting Emotions induced by Image Manipulation

Table 9.1 – Features used for predicting evoked emotions upon image manipulation.

Type	Dimension	Description
Color	1	A global factor measuring the colorfulness of an image [131]
	1	A global contrast factor of an image [132]
	48	Color histogram of image YCbCr colorspace (16-bin histogram for each channel)
Texture	22	Features from Gray-Level Co-occurrence Matrix (GLCM) <sup>6</sup> including the mean, variance, energy, entropy, etc.
Semantic	1	Number of people in an image
Manipulation	1	Manipulation method to be applied on an image

Figure 9.8 presents the overall framework of such a predictor, where the prediction targets include: the (i) valence score, (ii) arousal score and (iii) emotion keywords distribution. To train and evaluate the proposed predictor, we utilized the image dataset used in crowdsourcing experiment and collected emotions responses. In our dataset, there are 104 images, each manipulated by 7 different methods, resulted in 728 manipulated image samples. To train such a predictor, we extracted a set of features (color, texture and semantic information) from each original image, and then took another feature indicating the selected manipulation method. It finally resulted in a 74-dimension feature vector for each image. The detailed feature definitions are given in Table 9.1. To train and evaluate the proposed prediction model, we used the Scikit-learn machine learning library<sup>7</sup> [133] and experimented with four different methods: (i) Linear Regression, (ii) Support Vector Regression (SVR) with Radial basis function (RBF) kernel, (iii) Random Forest Regression (RFR) and (iv) a baseline method where the predicted value is simply the mean of the training set corresponding to each manipulation method. The baseline method is considered as a naïve approach where the image content is not taken into account while predicting evoked emotion. To avoid over-fitting, we conducted training and testing in 10-fold cross validation, with 90% images used for training and the rest for testing in each fold. As the emotion distribution is a vector instead of a scalar, its regression is considered as a multilabel regression problem. SVR in Scikit-learn does not directly support multilabel regression so we trained one classifier for each emotion category and normalized the seven predicted values in the evaluation phase so that they sum up to 1. Linear Regression and Random Forest Regression in Scikit-learn support multilabel regression by default and were therefore used directly. Two metrics are used to evaluate the regression performance: (i) mean squared error (MSE) and (ii) coefficient of determination (denoted  $R^2$ ). Particularly, for emotion distribution, the MSE value was obtained by averaging the squared differences between predicted distribution and groundtruth distribution over all the seven dimensions. For each prediction target (valence, arousal and emotion distribution), we tuned the parameters of the classifier to obtain as optimal result as possible. The final average results of cross validation tests are shown in Table 9.2.

<sup>7</sup><http://scikit-learn.org/>

Table 9.2 – Results of emotion prediction based on 10-fold cross validation.

Method	Valence		Arousal		Emotion Distribution	
	<i>MSE</i>	$R^2$	<i>MSE</i>	$R^2$	<i>MSE</i>	$R^2$
Baseline	1.26	-0.004	0.80	-0.011	0.0138	-0.011
Linear Regression	0.78	0.369	0.57	0.311	0.0092	0.325
SVR	0.51	0.589	0.37	0.529	0.0060	0.559
RFR	0.45	0.639	0.34	0.571	0.0056	0.585

From the results, one observes that Support Vector Regression and Random Forest Regression outperform the other two, while Random Forest Regression is slightly better than SVR. Random Forest Regression results in the highest  $R^2$  scores among all the methods, which are higher than 0.5 in predicting all the three targets: 0.639 for valence, 0.571 for arousal and 0.585 for emotion distribution. Such  $R^2$  scores indicate our predictor performs much better than random guess ( $R^2 \leq 0$ ) or the baseline method. When compared to the results of VA score prediction in [116], where the minimal MSE of 1.27 for valence and 0.82 for arousal were obtained, our prediction model for VA scores seems promising, although different datasets were used. With Random Forest Regression, we also checked the importance of features in decision making, which reveal that the manipulation method, number of people and energy component of GLCM are the three most important ones. This again indicates that image manipulation indeed influences image emotion in a high degree, and that high-level semantic features like the existence of people also have great impact on evoked emotion, in addition to other low-level image features such as color and texture.

## 9.5 Conclusion

This chapter investigates the influence of image manipulation on evoked emotions for different types of images. An image dataset was created by collecting different types of images from Instagram, and subjective experiments were conducted via crowdsourcing to examine subjects emotional responses on different images processed by different manipulations. Results of the user study show that certain image manipulations induce evoked emotions different from those experienced on original images. However, such manipulations do not always perform the same on different types of images. In other words, emotions induced by image manipulation depend not only on the applied manipulation but also on the image content. A further experiment was conducted in attempt to predict the emotions of a manipulated image given only its original version and the desired manipulation method. Using random forest regression based on a small set of image features, such a model reveals a promising accuracy. The results obtained from this study provide us with insights to design of advanced image emotion transformation systems that can recommend the type of manipulation to apply, based on the content of a picture and the desired emotion to express. This will serve as a future direction of our study.

## 10 Towards Dietary Management based on Image Analysis

Well-being is becoming a topic of great interest and an essential factor linked to improvements in the quality of life. Modern information technologies have brought a new dimension to this topic. It is now possible, thanks to various wearable devices (health bands, smart watches, smart clothes, etc.), to gather a wide range of information from subjects such as number of steps walked, heart rate, skin temperature, skin conductivity, transpiration, respiration, etc. and analyze this information in terms of the amount of calories spent, level of stress, duration and quality of sleep, etc. An accurate estimation of daily nutritional intake provides a useful solution for keeping healthy and to prevent diseases. However, it is not easy to assess the nutritional value of food and beverage consumed by subjects in an automatic and accurate way.

With the advancements of smart mobile phones equipped with high-resolution cameras, people capture and share a huge amount of photos every day, among which, food is one of the most popular subjects. In addition, recent years have seen a growing development of egocentric cameras or mobile capturing devices, such as GoPro<sup>1</sup>, Narrative Clip<sup>2</sup> and various smart watches with built-in camera, for lifelogging daily activities. All these provide us with the change to develop dietary assessment system based on multimedia techniques, e.g. food image analysis, though very challenging. An automatic image-based dietary assessment system usually follows the basic steps: food image detection, food item localization, recognition, segmentation, quantity or weight assessment, and finally caloric and nutritional value estimation [134]. In the last couple of years, advancements in image processing, machine learning and in particular deep learning techniques proved to be a boon for different image classification and recognition tasks, including for the problem of food image recognition. Researchers have been working on different aspects of a food recognition system, but we still lack a reliable system that can accurately estimate the nutritional value given an image containing some food. The reason lies in the difficulty to correctly recognize every fine-grained food item, as many different food items may

---

<sup>1</sup><https://gopro.com/>

<sup>2</sup><http://getnarrative.com/>

look extremely similar and are not even distinguishable to human eyes, for instance, beef vs. horse meat and rice vs. risotto. Moreover, in reality, a plate is usually full of highly mixed food, which makes the problem even more difficult to tackle. Therefore, we state that, it would be sufficient to recognize the general type of a food item in image and further to provide people with approximate information about their daily intake.

In this chapter, we address the problem of food image detection and recognition, using a deep learning approach, the convolutional neural network (CNN). We target on the initial steps of a complete dietary assessment procedure, namely, detecting images that contain food from users daily images and then classifying them into several major food categories. Therefore we report two sets of experiments: (i) food/non-food image classification, and (ii) food categorization, both based on fine-tuning a deep CNN, the GoogLeNet model, using a well-known framework deep learning, Caffe<sup>3</sup> [135]. To this end, we created two datasets from the existing food image datasets, social media and mobile devices for the two tasks respectively.

The rest of the chapter is structured as follows. Section 10.1 outlines the related work by other researchers including a short introduction of CNN, GoogLeNet model and fine-tuning technique. Section 10.2 describes the creation of food image datasets used for our experiments and Section 10.3 shows the experimental results on food/non-food classification and food category recognition. Section 10.4 introduces a prototype Android app for classifying food images and discusses privacy implications of such kind of system. Finally Section 10.5 concludes this chapter.

## 10.1 Prior Work

### 10.1.1 Food/Non-food Image Classification

The task of food/non-food image classification is to automatically detect the images that contain food items. It is an indispensable step for an automatic food analysis system where all daily images are treated as input. Classical approaches to image classification extract features such as interest point descriptors from scale-invariant feature transform (SIFT) [136, 137], pool the features into a vector representation e.g. bag of words [138] or Fisher Vectors [139] and then use a classification algorithm such as Support Vector Machine (SVM) to train a classifier. Various approaches have been proposed to solve the problem of food/non-food image classification. Kitamura et al. [140] applied SVM on image features consisting of color histograms, DCT coefficients and detected image patterns in food image detection and obtained an accuracy of 88%. [141] reports an automatic detector that finds circular dining plates in chronically recorded images or videos. As an important application, the method can be used to detect food intake events automatically by identifying dining plates in chronically recorded video acquired by a wearable device. Farinella et al. [142]

---

<sup>3</sup><http://caffe.berkeleyvision.org/>



employed three different image descriptors for food/non-food classification based on SVM: Bag of SIFT [136, 137], Pairwise Rotation Invariant Co-Occurrence Local Binary Pattern (PRICoLBP) [143] and Bag of Textons [144]. The best result obtained on the UNICT-FD889 dataset [145] is 94.44%. Recently, the convolutional neural network (CNN) [146] offers a state-of-the-art technique for many general image classification problems and is therefore used in the problem of food image classification. Kagaya et al. [147] applied CNN in food/non-food classification and achieved significant results with a high accuracy of 93.8%. Then, in their study [148], the accuracy of food detection is improved to 99.1%. Compared to previous studies using conventional machine learning approaches, deep-based approaches like CNN provide significantly better performance.

### 10.1.2 Food Image Recognition

For dietary assessment, system should be able to also find out what food items are in an image, their locations, as well as their amount. Therefore, another essential step other than food/non-food classification is to recognize the food item in an image. Most solutions in food recognition assume only one food item present in an image. Thus, food recognition can be solved as a multiclass classification problem. Researchers have been working on food recognition using conventional approaches based on classical image feature descriptors and machine learning for many years. Joutou et al. [149] created a private Japanese food dataset with 50 classes. They proposed a Multiple Kernel Learning (MKL) method using combined features including SIFT-based bag-of-features, color histogram and Gabor Texture features. An accuracy of 61.3% on their dataset was achieved. A follow-up study by Hoashi et al. [150] achieved an accuracy of 62.5% using the same method on an extended dataset of 85 food classes. Chen et al. [151] created the Pittsburgh food database which contained 101 classes of American fast food images taken in a controlled environment. Yang et al. [152] defined eight basic food materials and learned spatial relationships between these ingredients in a food image using pairwise features. They achieved a classification accuracy of 28.2% on 61 food categories which was a subset of Pittsburgh dataset [151]. Bettadapura et al. [153] used combined 6-feature descriptors (2 color-based and 4 SIFT-based) and SMK-MKL Sequential Minimal Optimization to train an SVM classifier. They experimented on a dataset consisting of 3750 food images of 75 categories (50 images per category) and reported an accuracy of 63.33% on their test dataset. Interestingly, they incorporated the geological information of where the food picture was taken so that they could get the information about the restaurant and then downloaded the menu online. An assumption of their work is that the food image must be one of the items in the menu. Rahmana et al. [154] presented a new method for generating scale and/or rotation invariant global texture features using the output of Gabor filter banks, which provides a good accuracy of food classification for a mobile phone based dietary assessment system. The top-5 accuracy they achieved was almost 100%. However, the experiment was conducted on a special image dataset of only 209 food images created with controlled environment. He et al. [155] investigated different features

and their combinations for food image analysis and a classification approach based on k-nearest neighbors (k-NN) and vocabulary trees. The experimental results indicate that a combination of three features, Dominant Color Descriptor (DCD) [156, 157], Multi-scale Dense SIFT (MDSIFT) [158] and Scalable Color Descriptor (SCD) [156], provides the best performance on food recognition. Bossard et al. [159] created an image dataset called Food-101, which contains 101 types of food images. They presented a method based on Random Forests to mine discriminative visual components and could efficiently classify with an accuracy rate of 50.8%.

In recent years, CNN has been widely used in food recognition and provides better performance compared to conventional approaches. Bossard et al. [159] trained a deep CNN from scratch on Food-101 dataset using the architecture of AlexNet model (proposed by Krizhevsky et al. [160]) and achieved 56.4% top-1 accuracy. In [147], Kagaya et al. also trained a CNN for food recognition and experimental results revealed that their method outperformed all the other baseline classical approaches by achieving an average accuracy of 73.7% for 10 classes. Kawano et al. [161] used CNN as a feature extractor and achieved state-of-the-art best accuracy of 72.3% on the UEC-FOOD-100 [162] dataset, which contains 100 classes of Japanese food. They used the pre-trained AlexNet model as a feature extractor and integrated both CNN features and Fisher Vector [139] encoded SIFT and color features. Yanai et al. [163] fine-tuned the AlexNet model and achieved the best results on public food datasets so far, with top-1 accuracy of 78.8% for UEC-FOOD-100 dataset and 67.6% for UEC-FOOD-256 [164] (another Japanese food image dataset with 256 classes). Their works showed that the recognition performance on small image datasets like UEC-FOOD-256 and UEC-FOOD-100 (both of which contained 100 images for each class) can be boosted by fine-tuning the CNN network which was pre-trained on a large dataset of similar objects. Myers et al. [165] presented the Im2Calories system for food recognition which extensively used CNN-based approaches. The architecture of GoogLeNet [166] was used in their work and a pre-trained model was fine-tuned on Food-101. The resulted model obtained a top-1 accuracy of 79% on Food-101 test set.

### 10.1.3 Convolutional Neural Network

Over the last few years, due to the advancements in the deep learning, especially in the convolutional neural networks (CNN), the accuracy of image classification has been increased drastically. This is not only because larger datasets but also new algorithms and improved deeper architectures [166]. CNN is also known as LeNet due to its inventor [167]. It mainly comprises convolutional layers, pooling layers and sub-sampling layers followed by fully-connected layers. The very first architecture of CNN [146] takes as input an image and applies convolution followed by sub-sampling. After two such computations, the data is fed into the fully connected neural network, where it performs the classification task [146]. The main advantage of CNN is the ability to learn the high-level efficient features and in addition to that, it is robust against small rotations and shifts. Significant progress

has been made on this basic design of CNN and it has been extended by increasing the number of layers [168], size of layers [169] and better activation function (e.g. ReLU [170]) to yield the best results on various challenges related to object classification, recognition and computer vision.

In this study, we employ the GoogLeNet [166], a deep network consists of 22 layers constructed based on the architecture of CNN. GoogLeNet is an efficient deep neural network architecture, which has a new level of organization called *Inception Module*. Such module basically acts as multiple convolution filter inputs, which are processed on the same input. It also does pooling at the same time and all the output results are then concatenated. Fully-connected layers are being replaced with parallel convolutions that operate on the same input layer. An Inception Module in GoogLeNet consists of several convolutions and a max pooling operation and there are nine such modules in a GoogLeNet architecture. The  $1\times 1$  convolutions at the bottom of the module reduce the number of inputs and hence decrease the computation cost dramatically. It also captures the correlated features of an input image in the same region. Where as, image patterns are responded by  $3\times 3$  and  $5\times 5$  convolutions at larger scales. Feature maps which are being produced by all the convolutions are concatenated to form the output [166]. Using Inception Module allows the network to take advantage of multi-level feature extraction from each input. It allows increasing number of units significantly without an uncontrolled blow-up in computational complexity. The resulted networks are usually 2-3 times faster than similarly networks without inception architecture.

#### 10.1.4 Transfer Learning and Fine-Tuning

In this study, we employ an important machine learning strategy, named transfer learning [171]. Common machine learning algorithms usually address isolated tasks. While, transfer learning attempts to change this concept by developing methods to transfer knowledge learned in one or more source tasks and use it to improve learning in a related target task [171]. The goal of transfer learning is to improve learning in the target task by leveraging knowledge from the source task. Pratt is the first who came up with the concept of transfer learning and proposed the discriminability-based transfer (DBT) algorithm [172] in 1993. DBT uses an information measure to estimate the utility of hyperplanes defined by source weights in the target network, and rescales transferred weight magnitudes accordingly. Research on transfer learning has attracted more and more attention since 1995 in different names: learning to learn, knowledge transfer, inductive transfer, multi-task learning, knowledge consolidation, context-sensitive learning, knowledge-based inductive bias, meta learning, and incremental/cumulative learning [173, 174, 175].

In practice, we don't usually train an entire deep CNN from scratch with random initialization. This is because it is relatively rare to have a dataset of sufficient size that is required for the depth of network required. For instance, the GoogLeNet was initially

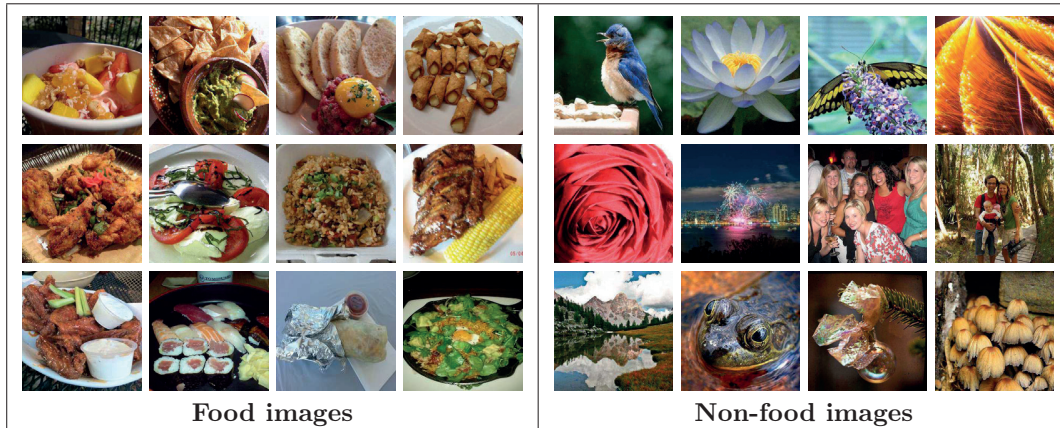


Figure 10.1 – Image samples of Food-5K dataset.

trained on ImageNet [176], a dataset containing over 1 million natural images categorized in 1000 classes. Instead, it is common to use the weights of an existing CNN pre-trained on a very large dataset (e.g. ImageNet), as either an initialization (fine-tuning) or a fixed feature extractor for the task of interest. The fine-tuning strategy [177] fine-tunes the weights of a pre-trained CNN by continuing the backpropagation. It is possible to fine-tune all the layers of a CNN or to some higher-level portion of the network while keeping some of the earlier layers fixed (due to overfitting concerns). This is motivated by the fact that the earlier features of a CNN usually contain more generic features (e.g. edge detectors or color blob detectors) that should be useful to many tasks but later layers of the DCNN becomes progressively more specific to the details of the classes contained in the dataset. In this study, we also applied fine-tuning on a pre-trained GoogLeNet model to build our food image classifiers.

## 10.2 Datasets

First, we created two image datasets<sup>4</sup>, named **Food-5K** and **Food-11**, used for the experiments on food/non-food classification and category recognition respectively. Both datasets are split into three subsets, for the purpose of training, validation and evaluation respectively. In addition, another Instagram Food/Non-Food Dataset (**IFD**) created by [148] was used in our experiments to evaluate the performance of our model on food/non-food classification. Details of the three datasets are given as follows.

**Food-5K** dataset contains 2'500 food images and 2'500 non-food images, resulting in a total of 5'000 images. The food images were selected from existing and publicly available food image datasets, including Food-101 [159], UEC-FOOD-100 [162] and UEC-FOOD-256 [164]. The food images were selected in such a way that they could cover

<sup>4</sup>The datasets are publicly accessible in <http://mmspg.epfl.ch/food-image-datasets>.



Figure 10.2 – Image samples of Food-11 dataset.

a wide variety of food items. This may help to train a strong classifier that can detect food images with a wide variety. In addition, those images in which food is not even the main portion of image content are also considered as food image. For non-food images, we randomly selected 2'500 from existing image datasets consisting of general non-food objects or humans. These datasets include Caltech101 [178], Caltech256 [179], the Images of Groups of People [180] and Emotion6 [122]. We tried to cover a wide range of contents in the non-food images and included some non-food images visually similar to food, thus increasing the difficulty of classification task. Every image was visually inspected by us such that it is distinguishable by a human observer in terms of its belongingness to one of the two classes: food and non-food. For the training phase, we used 3'000 images (1'500 for food and 1'500 for non-food). The rest of the dataset was equally divided into two subsets, for validation and evaluation respectively. Figure 10.1 shows some examples of food and non-food images in Food-5K dataset.

**Food-11** dataset consists of 16'643 images grouped into 11 categories, which basically cover the major types of food that people consume in daily life. We defined the food categories by adopting and modifying the major food groups defined by United States Department of Agriculture (USDA) [181]. The 11 categories are: *Bread*, *Dairy products*, *Dessert*, *Egg*, *Fried food*, *Meat*, *Noodles/Pasta*, *Rice*, *Seafood*, *Soup* and *Vegetable/Fruit*. The dataset was mainly collected from existing food image datasets including Food-101 [159], UEC-FOOD-100 [162] and UEC-FOOD-256 [164]. For certain categories (*Dairy products* and *Vegetable/Fruit*), we downloaded images from two social media sites, i.e. Flickr and Instagram. For each food category, we tried to include different food items in

Table 10.1 – Categories, example items and number of images in each subset of Food-11.

Category	Example items	Train	Val.	Eval.
Bread	Bread, burger, pizza, pancakes, etc.	994	362	368
Dairy products	Milk, yogurt, cheese, butter, etc.	429	144	148
Dessert	Cakes, ice cream, cookies, chocolates, etc.	1500	500	500
Egg	Boiled and fried eggs, and omelette.	986	327	335
Fried food	French fries, spring rolls, fried calamari, etc.	848	326	287
Meat	Raw or cooked beef, pork, chicken, duck, etc.	1325	449	432
Noodles/Pasta	Flour/rice noodle, ramen, and spaghetti pasta.	440	147	147
Rice	Boiled and fried rice.	280	96	96
Seafood	Fish, shellfish, and shrimp; raw or cooked.	855	347	303
Soup	Various kinds of soup.	1500	500	500
Vegetable/Fruit	Fresh or cooked vegetables, salad, and fruits.	709	232	231
Total		9866	3430	3347

order to increase the difficulty of recognition. Apart from this, only those images whose main content is food of that particular category were selected. The concrete example food items in each category and the number of images for each subset are listed in Table 10.1. Figure 10.2 shows example food images of the 11 categories.

**IFD** dataset was built searching results for #tag “food” in Instagram, being manually annotated with food and non-food labels. The dataset consists of 4’230 food images and 5’428 non-food images. In [148], the food/non-food classification experiments conducted on IFD dataset resulted in a maximum accuracy of 95.1%. We use this dataset in our experiments to evaluate the performance of our model and compare with the classification results in [148].

### 10.3 Experiments and Analysis

This section describes the experiments on food/non-food classification and food category recognition carried out using the datasets we created. In the experiments, we use Caffe [135], one of the most popular frameworks for deep convolution neural network. A pre-trained GoogLeNet model is applied and fine-tuned using our dataset for both food/non-food classification and food categorization. In particular we provide details on how the refinement of the model was achieved.

#### 10.3.1 Food/Non-food Classification

Food/Non-food classification, or food image detection, is one of the initial and important steps for image-based dietary assessment. To classify food and non-food images, we use a pre-trained GoogLeNet model<sup>5</sup> and fine-tuned it using the training subset of Food-

---

<sup>5</sup>[https://github.com/BVLC/caffe/tree/master/models/bvlc\\_googlenet](https://github.com/BVLC/caffe/tree/master/models/bvlc_googlenet)

Table 10.2 – Accuracy of food/non-food image classification on evaluation set of Food-5K for the two fine-tuning configurations.

Iteration #	3,000	4,000	5,000	6,000	7,000	8,000	9,000	10,000
Last 2 layers	0.953	<b>0.983</b>	0.972	0.970	0.979	0.980	0.978	0.979
Last 6 layers	<b>0.987</b>	0.976	<b>0.974</b>	<b>0.975</b>	<b>0.992</b>	<b>0.981</b>	<b>0.983</b>	<b>0.982</b>

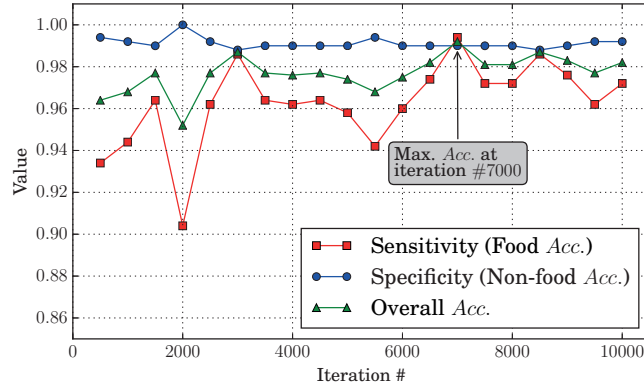


Figure 10.3 – Results of food/non-food classification obtained on evaluation set of Food-5K.

5K dataset. Fine-tuning process takes a pre-trained model, adapts the architecture, and resumes training from the already learned model weights. When fine-tuning a pre-trained GoogLeNet model, we can choose the layers the parameters of which should be updated. Based on the existing GoogLeNet, we set the base learning rate to 0.01 using a polynomial learning rate policy. Besides, the maximum number of iterations is set to 10000. Then we set up two configurations to fine-tune the GoogLeNet model, with one only updating the parameters of the last two layers and the other for the last six layers. The overall classification accuracies (on evaluation set) of the two configurations for different iterations are shown in Table 10.2. In most cases especially for higher number of iterations, higher accuracy is achieved on the second setup of fine-tuning i.e. the last six layers of GoogLeNet model. Therefore, we kept using the setup of fine-tuning the last six layers in the remaining experiments. Figure 10.3 shows the detailed results of food/non-food classification on the evaluation subset of Food-5K by fine-tuning the last six layers of GoogLeNet. In the results, the sensitivity, or true positive rate, indicates the rate of correctly detected food images. While, the specificity, or true negative rate, refers to the rate of correctly detected non-food images. From Figure 10.3, a maximum accuracy rate of 99.2% was achieved on evaluation dataset at iteration #7000, with sensitivity and specificity of 99.4% and 99.0% respectively. This means only 8 images out of the whole evaluation set (1000 images) are incorrectly classified. Figure 10.5 shows the 8 incorrectly classified samples for iteration #7000. Some non-food images classified as food are highly similar to food images and those food images classified as non-food images are either ambiguous or containing a very small region of food. Figure 10.4(a) shows the confusion matrix of food/non-food classification on our own dataset Food-5K.

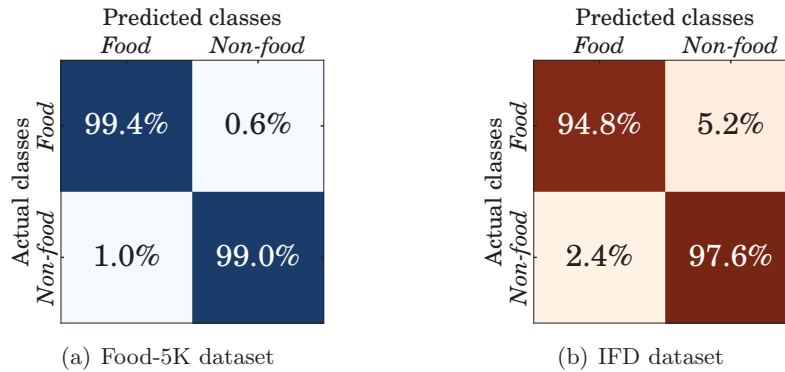


Figure 10.4 – Confusion matrix of food/non-food classification results on two different image datasets: Food-5K and IFD.

To further evaluate the performance of our fine-tuned classifier on food/non-food classification, we ran our classifier on the other two datasets: Food-11 dataset created by us, and Instagram Food/Non-Food Dataset (IFD) by Kagaya et al. [148]. For both datasets, we tested our classifier at iteration #7000. For Food-11 dataset, we ran the food/non-food classifier on all the 16'643 food images and 16'127 of them were correctly detected as food images, resulting in a detection rate of 96.9%. Note that there are only food images in Food-11 dataset so the accuracy is just the rate of correctly detected food images. For IFD dataset [148], we evaluated our classifier on randomly selected 500 food and 500 non-food images. The classification result is shown as confusion matrix in Figure 10.4(b). Among all the 500 food images, 474 (94.8%) were correctly classified as food, while 488 (97.6%) out of 500 non-food images were correctly classified as non-food. This resulted in an overall accuracy of 96.4%, which is slightly higher than the maximum accuracy of 95.1% obtained in [148]. In this regard, the performance of our classifier obtained on fine-tuning a GoogLeNet is promising.

### 10.3.2 Food Image Categorization

In the second experiment, we used Food-11 dataset to train and evaluate a CNN classifier on food image categorization. The food images in Food-11 have been categorized into 11 classes and Table 10.1 shows the number of images in each category for training, validation and evaluation. The task for the second experiment is to classify each food image into one of the 11 categories. For this purpose, we again applied the pre-trained GoogLeNet model<sup>6</sup> and fine-tuned its last six layers on the training set of Food-11. We used a base learning rate of 0.001 with the same polynomial policy. To evaluate the performance of categorization, we use three metrics: (i) overall accuracy  $Acc.$ , (ii) F-measure  $F_1$  [182], and (iii) Cohen's kappa coefficient  $\kappa$  [183]. Specially, the Cohen's kappa coefficient is

<sup>6</sup>[https://github.com/BVLC/caffe/tree/master/models/bvlc\\_googlenet](https://github.com/BVLC/caffe/tree/master/models/bvlc_googlenet)





Figure 10.5 – Misclassified food and non-food images in Food-5K dataset.

a numerical evaluation of inter-rater agreement which takes into account not only the observed classification accuracy but also the accuracy that any random classifier would be expected to achieve, namely, random accuracy. It is especially useful in evaluation of classification when the quantities of images in different classes are unbalanced.

Figure 10.6 shows the overall accuracy, F-measure and Cohen’s kappa coefficient of food categorization on the evaluation subset of Food-11. The maximum accuracy of 83.5% is achieved at iteration #4100, where we also obtain the maximum values of F-measure and kappa coefficient of 0.911 and 0.816 respectively. The high value of kappa coefficient (0.816) indicates that the trained classifier performs significantly better than any random classifier. Due to time constraints, we had to stop evaluating the results on the evaluation dataset after iteration #5000, as the accuracy on the validation dataset did not show any significant improvement. The confusion matrix of the recognition result at iteration #4100 is shown in Figure 10.7. Among all the classes, *Noodles/Pasta*, *Rice* and *Soup* result in the best accuracies, higher than 95%. This is because the food images in these categories have their respective common characteristics in either shape or color and are therefore easier to be identified. However, one notices that some types of food images are error-prone, e.g. *Bread*, *Egg* and *Meat*, the accuracies of which are lower than 70%. Those types of images are usually of highly mixed food items in our dataset, namely high intra-class variation. For instance, category *Egg* contains boiled egg, fried egg and omelette, which are highly different in appearance. Besides, many of those images have the main food mixed with other food items, e.g. meat with salad. Interestingly, one also observes that *Dessert* and *Soup* are the two target classes to which other food images are mostly misclassified. In 7 classes (*Bread*, *Dairy*, *Egg*, *Seafood*, *Meat*, *Fried food* and

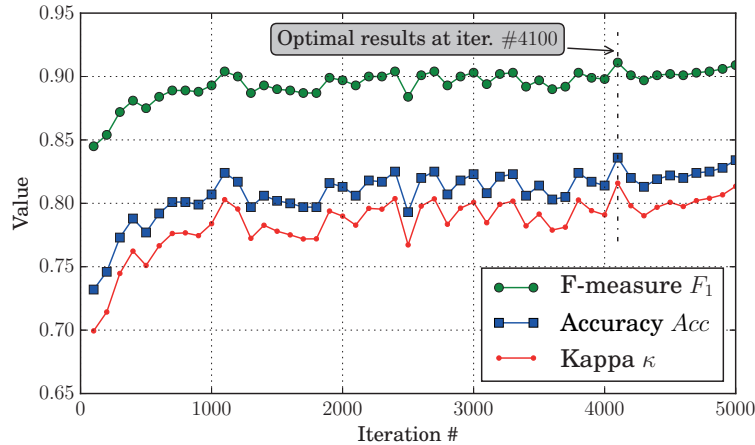


Figure 10.6 – Performance of food categorization on evaluation set of Food-11 dataset.

*Vegetable/Fruit*), more than 5% of their evaluation set are incorrectly classified as *Dessert*. This might be because *Dessert* is the category that has the most mixed items in our dataset, and that many of them could be visually similar to other food. Besides, more than 4% of images in *Bread*, *Dessert*, *Meat* and *Seafood* were misclassified as *Soup*. By checking some of images misclassified as soup, we found most of them have round-shaped elements such as plate or round bread. In our dataset, most *Soup* images also have the similar round-shaped plates or containers. According to the confusion matrix in Figure 10.7, we list the top 10 incorrectly classified class pairs and show two example images for each in Figure A.9 in Appendix A. By observing the incorrectly classified images, we find that incorrect classifications are mainly due to the following two reasons: (i) Images within a single food category may contain various types of food items in very different appearances, or an image may contain mixed food items, known as high intra-class variation; (ii) Images across different classes may share similar appearance, shape or color, namely high inter-class similarity. Considering the fact that each image category in Food-11 dataset contains different food items with certain varieties, and that the size of our dataset is not considerably large, the results obtained are promising.

## 10.4 Discussions

**Prototype Android Food Recognizer** Using the classifiers trained in our experiments, we developed a prototype Android mobile app demonstrating the food image classification and categorization processes. The application takes as input an image captured from camera or selected from photo gallery, uploads the image to a dedicated server, and triggers the classification core on the server. The server first executes the food/non-food classifier on the image and determines whether or not the image is a food image. If the image is not detected as food image, the application returns “non-food” to user. If yes, the second classifier for food categorization is executed such that the

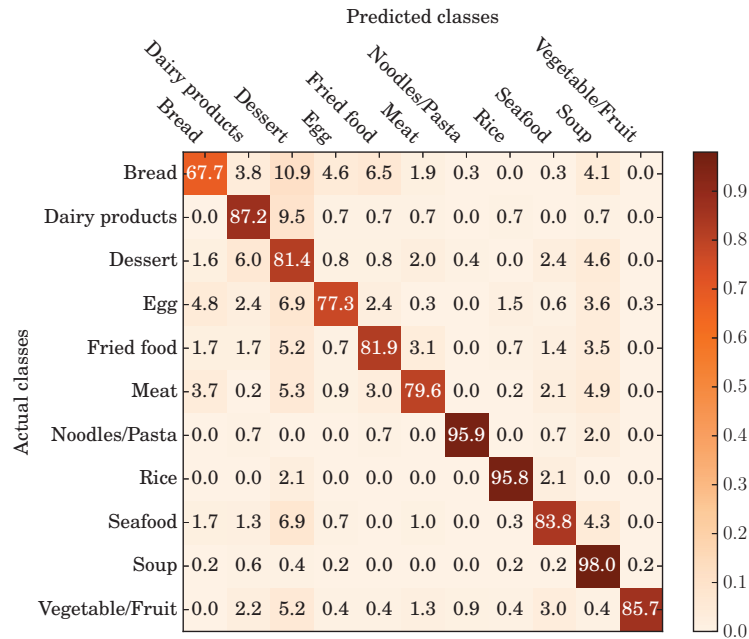


Figure 10.7 – Confusion matrix of food recognition. Values of the matrix are in percentage.

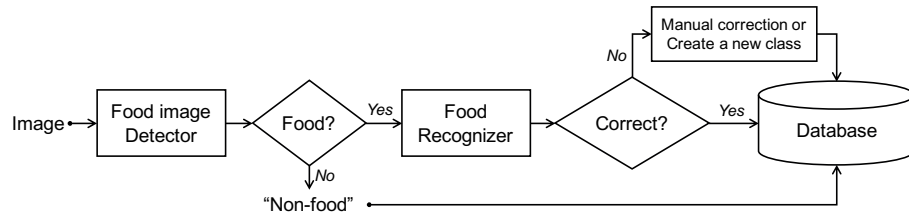


Figure 10.8 – Workflow of a food classification system for dietary assessment.

user will receive a predicted label of the food image. The user can then manually check if the classification result is correct, and if not, provide the correct label by choosing from a predefined list of food categories or creating a new food class. Therefore, such an application also offers a potential to collect new food images and fine-grained classes from users. The workflow of such a food image classification system and several screenshots of the mobile application are shown in Figure 10.8 and Figure 10.9 respectively.

**Privacy Implications** Several issues remain with such a dietary assessment system, among which privacy implication is significant. Using wearable or egoistic camera may not only reveal private information about the user him/herself, but also accidentally violate the privacy of other people being shot by an “invisible” device. In practice, such privacy implication should be properly addressed. Possible solutions include secure protection of image files or removal of sensitive parts from image with the privacy protection algorithms

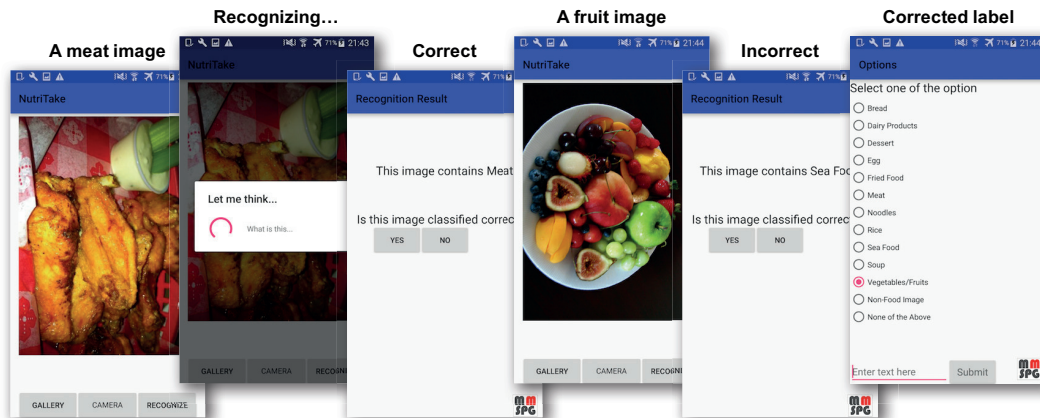


Figure 10.9 – Screenshots of prototype Android app for food image classification.

proposed in the thesis.

## 10.5 Conclusion

In this chapter, we explore applying a pre-trained GoogLeNet for the tasks of food/non-food image classification and food image categorization. To this end, we first constructed two image datasets from publicly available image datasets, social media and our mobile cameras. We then trained two classifiers for the two tasks respectively by fine-tuning the last six layers of a GoogLeNet model. Evaluation results of the classifiers performed on the evaluation sets show the best accuracy of 99.2% on food/non-food image classification and 83.6% on food categorization respectively. Based on the trained classifiers, we developed a prototype Android application demonstrating food image classification and categorization processes. The mobile application communicates with a dedicated server where all images are uploaded on and classified. The classified result is returned to user so that the user can check its correctness and afterwards take measures to either accept the result or manually correct it; in this regard, such type of application is also seen as a food image collector to enrich the quantity and categories of an existing food image dataset.

For future work, we aim at integrating contextual information to improve the accuracy of food classification and compare it with other different architectures such as AlexNet [160], VGG [184] or ResNet [185]. We will also work on the estimation of food items quantity and weight in order to finally estimate their nutritional value. Last but not least, we will take into account privacy implications of using such an approach for dietary assessment, as a great amount of sensitive information other than food may be released from those images. We plan to investigate efficient solutions to enable both privacy and usability, namely, privacy-preserving dietary management.

# 11 Summary and Future Work

## 11.1 Thesis Summary

In this thesis, we investigate novel solutions to protect image privacy, with a particular emphasis on the online photo sharing environment. To this end, we propose not only image encoding algorithms but also architecture designs to enhance visual privacy in photo sharing. Moreover, we explore the potentials and additional impacts of using daily images captured, edited or shared by people in other three relevant applications beyond privacy.

First, we propose and study two image encoding algorithms to protect visual information in image, within a Secure JPEG framework. The first method scrambles a JPEG image by randomly changing the signs of the quantized DCT coefficients corresponding to predefined image regions. The second method, named JPEG Transmorphing, allows one to protect arbitrary image regions by applying any type of regional obfuscation, while secretly preserving the original image regions in application segments of the obfuscated JPEG image. Both algorithms are JPEG backward compatible, meaning that the protected image is readable by any legacy JPEG decoder in its visually protected form. Only with a special decoder and a correct secret key, the original image can be recovered. Both objective and subjective experiments were conducted to evaluate the performances of the two protection algorithms with respect to their storage overhead, reconstruction quality, privacy protection capability or pleasantness.

Particularly, the JPEG Transmorphing, designed with a completely new philosophy from any existing privacy protection scheme, presents a high flexibility and usability compared to the other distortion-based schemes. JPEG Transmorphing allows one to apply any personalized obfuscations in image regions, while still preserving the reversibility in the protected image, even with lossy transformations applied. We conducted a subject experiment to inspect the privacy protection capability of several regional obfuscation methods against person recognition. Experimental results indicate that a

regional obfuscation of image visual information can well preserve the privacy of the obfuscated person, unless the “attacker” has access to considerable prior knowledge about the protected person that highly relates to the context information disclosed from the unprotected regions. Among all obfuscations under evaluation, visual masking shows the optimal performance compared to the others, such as JPEG Scrambling, P3 and blurring and pixelation. Thanks to the proposed JPEG Transmorphing algorithm, any type of regional visual masking can be applied in a reversible manner. Therefore, the JPEG Transmorphing method could also provide a better level of pleasantness than other distortion-based approach from both observer and user’s perspective. This fact has been proven in another subjective experiment based on crowdsourcing. To conclude, JPEG Transmorphing proved to satisfy all the requirements expected for an ideal privacy protection scheme outlined in the beginning of the thesis (Chapter 1).

In the second part, we investigate the design of two architectures for privacy-preserving photo sharing. In the first architecture, named ProShare, a public key infrastructure (PKI) integrated with a ciphertext-policy attribute-based encryption (CP-ABE) is applied to enable an efficient and secure sharing of images protected by Secure JPEG algorithms. In ProShare, a photo is securely encoded by a Secure JPEG protection algorithm with a secret key and then securely kept on an untrusted service (server, cloud, etc.). Meanwhile, the secret key is encrypted by CP-ABE with a user-defined access policy. With the help of the PKI, users can share ABE private keys between each other and only those privileged users who possess ABE private key associated with right attributes are able to recover the original image. We implemented and demonstrate the correct and efficient functioning of the ProShare architecture based on both iOS and Android mobile platforms.

The second architecture is named ProShare S, designed in a completely different mechanism compared to ProShare. In ProShare S, a photo sharing service provider helps users make photo sharing decisions automatically based on their past decisions made in different contexts. The sharing service takes into account not only the content of a user-posted image, but also the context information about the image capture and a prospective requester. The decision making is achieved by a classifier analyzing the above semantic and contextual information. To validate the ProShare S architecture, we conducted a user study on 23 users along with extensive performance evaluations. Experimental results reveal a promising accuracy of the prediction model with a minimum burden from users.

As the last part of the thesis, we research into three relevant topics in regard to daily photos captured or shared by people, but beyond their privacy implications. In the first study, we adopt the idea from JPEG Transmorphing and propose an animated JPEG file format, named aJPEG. aJPEG “simulates” the GIF, but preserves its animated frames as APPn markers in a “cover” JPEG image, which encodes a selected default frame. The aJPEG format proved to offer smaller file size and higher image quality than conventional GIF, and therefore could serve as a better alternative to GIF. In the second

study, we attempt to understand the influence of popular image manipulations applied in online photo sharing on evoked emotions of photo observers. The study reveals the fact that image manipulations indeed impact people's evoked emotion, but such impact is also dependent on image content. By learning from image features such as color and texture, we train and evaluate a regressor that can accurately predict the emotions of a manipulated image given input as the original image and a desired manipulation. In the last study, we target on the problem of dietary management using daily photos captured by people. To this end, we employ a deep CNN model, the GoogLeNet, to perform automatic food image detection and categorization. Promising results obtained in both tasks provide us with significant insights in design of automatic dietary assessment system based on multimedia techniques, e.g. image analysis.

## 11.2 Future Directions

Several limitations of our studies still remain. In the proposed JPEG Transmorphing (Chapter 5), we only focus on the encoding method itself by assuming knowing the sensitive image regions to protect or the existence of techniques that can automatically detect and identify private ROIs. At most, we apply a simple face detector to detect face regions in image. Yet, identification of the privacy-sensitive regions in image is a research topic that needs particular efforts. Substantial research has already been devoted in it using different computer vision and pattern recognition techniques including the recent deep learning networks [43, 41, 42]. We envision a promising future of private ROI detection in image using deep learning techniques in combination of various context information, e.g. time, location and device.

When applying JPEG Transmorphing to protect image privacy, we only consider visual information in image, which might be inadequate as metadata associated in image may also reveal privacy. However, it is not difficult to achieve the protection of image metadata using a similar approach as JPEG Transmorphing by means of secretly hiding metadata along with Transmorphing data in image. In addition, we consider only single level information protection and sharing, in both protection algorithms and photo sharing architectures. It would be interesting and challenging to achieve multi-party photo sharing using the proposed approaches, namely, disclosing different versions of the same image content to different entities depending on their privileges.

Recent years have also seen a boom on video or animation sharing on social networks. Most common requirements for privacy protection of image content will still apply for video content, such as low overhead, low complexity and backward compatibility with existing video coding algorithms. However, due to different natures of video and animation content and coding schemes applied, they may raise new requirements and challenges. In the next step, we also plan to extend our encoding algorithms and architecture to enable the privacy protection for video content.

In the study of context-dependent privacy-aware photo sharing architecture ProShare S (Chapter 7), most image semantic features are manually annotated by our subjects, which is impractical when being applied in practice. Also, results of the study are generated on the data of only 20 effective subjects and they were put in a hypothetical environment and provided their sharing decisions upon the artificial questions we asked. These are mainly due to the lack of access and control of a popular social network. Therefore, integrating necessary automatic feature extractions and putting greater number of users in a realistic social networking environment will serve as our future work. This will further help us understand users photo sharing behaviors with respect to privacy concerns. Furthermore, we will investigate more sophisticated machine learning and even deep learning approaches to understand the privacy values of user-posted media content, and take into account more contextual information, to build more accurate and secure privacy protection mechanism in online photo sharing.

Last but not the least, privacy is a highly complex issue in the scenario of social media, due to the fact that information is being created and shared extensively within a large user graph. Our current study mainly focuses on the privacy issues associated with the media content itself, without taking into account the implications from the complex user connections in a networked environment. We call it content-centric approach. We believe not only the media content itself but also other information about the user (friends, profile, comments, or even friends' content etc.) may compromise user's privacy. Therefore, future research for privacy protection can be carried out in a user-centric approach by taking into account both media content and other information associated to user.



## A Screenshots and Supplied Images

## Appendix A. Screenshots and Supplied Images







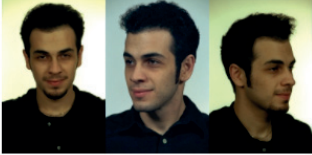


**Instructions**

In this HIT, you are asked to **recognize the person in an image from 9 candidate people**. The image is either obfuscated by certain image processing filter or kept in its original form without obfuscation. Please follow the below steps to conduct the HIT:


1. Before starting the HIT, you have to **view the reference photos of the 9 candidate people**, shown in the section "**Reference Photos**". Try to memorise their faces. You can always come back to check those reference photos when answering questions.
2. Then, **try to recognize the person in the section "TASK" from the 9 reference people**.
3. If you really don't know who is in the picture, select "**I really don't know.**"

We will check the trustworthiness of your answers. Please provide your answers carefully. Thank you very much for your effort.

**Reference Photos**

<b>Person1:</b> 	<b>Person2:</b> 	<b>Person3:</b> 
<b>Person4:</b> 	<b>Person5:</b> 	<b>Person6:</b> 
<b>Person7:</b> 	<b>Person8:</b> 	<b>Person9:</b> 

**TASK:**



Who is in the picture?

I really don't know.

<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	<input type="radio"/>						<input type="radio"/>

Figure A.1 – Screenshot of an HIT on AMT for subjective experiment on face recognition (Chapter 4).

142

**Instructions**

The shown picture contains a car or truck with a license plate. The picture is in either original clear form, or visually obfuscated form by a certain image processing method. Your task is to try to find and identify the license plate in the picture, and write down the code of the license plate in the text field. Please note:

- Write down only the number and letter in UPPERCASE. Ignore any special character, logo and space.
- You can try your best to guess.
- If you really have no clue, simply put "0" (zero).
- Below you have an example:

For the picture on the left, you should write:

If you cannot recognize the license plate, write:

Figure A.2 – Screenshot of an HIT on AMT for subjective experiment on license plate recognition (Chapter 4).

## Appendix A. Screenshots and Supplied Images

---

### John:



### Leon:



### Eric:



### David:



### Mark:



### Carl:



Figure A.3 – Evaluation images of the six identities in subjective evaluation of different privacy protection methods (Chapter 5).

**Instructions**

**Suppose you are in the scenario of an online photo sharing platform, e.g. Facebook or Instagram. Users may apply privacy protection in their photos against unauthorized users (such as strangers or public). While, only authorized users (e.g. friends) will be able to see the original picture thanks to an access control mechanism.**


In this HIT, you need to provide your personal opinion on certain image privacy protection method applied in picture. In the HIT, a picture is shown to you. A face in the picture is protected by a visual privacy protection method. You need to answer two questions on the applied privacy protection method:

1. Your perceived feeling/emotion when seeing the picture protected in such a way. **Assume the picture belongs to a person who wants to protect his/her face privacy from public and you fully understand this fact because the person is not your friend.**
2. Your personal preference to use the particular method applied in the picture to protect your own photo privacy when sharing photos online.

We will check the trustworthiness of your answer. Careless answers will be rejected. So please conduct the HIT carefully. Thanks!

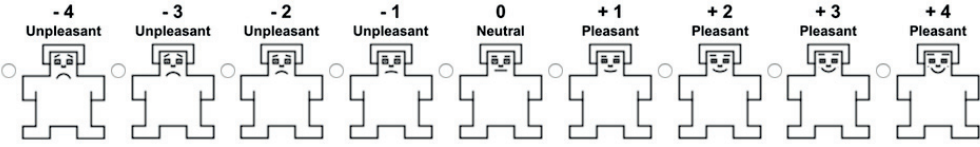
---

**Start the HIT:**



**Q1. What is your feeling when seeing this picture with a person's face protected in such a way? Assume this is a picture from another person who wants to protect his/her face privacy from public, and you could understand this fact because the person is not your friend. Select the picture that best describes your feeling or emotion.**

-4 Unpleasant   -3 Unpleasant   -2 Unpleasant   -1 Unpleasant   0 Neutral   +1 Pleasant   +2 Pleasant   +3 Pleasant   +4 Pleasant



**Q2. Do you like to use the particular method in the picture to protect your own face privacy against public access to your photos in online social network?**

Dislike   Neutral   Like




Figure A.4 – Screenshot of an HIT on AMT for subjective experiment on pleasantness of visual privacy protection methods (Chapter 5).

## Appendix A. Screenshots and Supplied Images

Figure A.5 – Application screenshots.

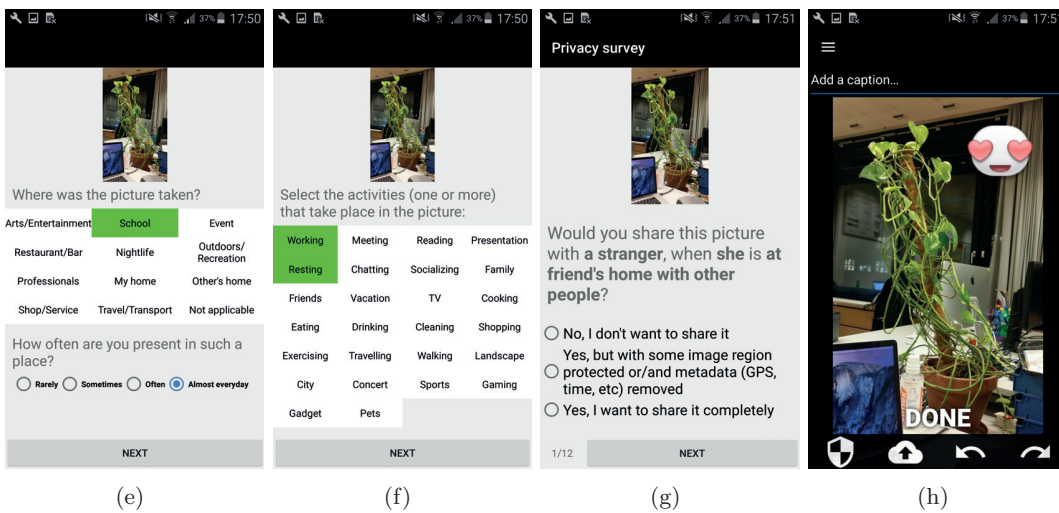
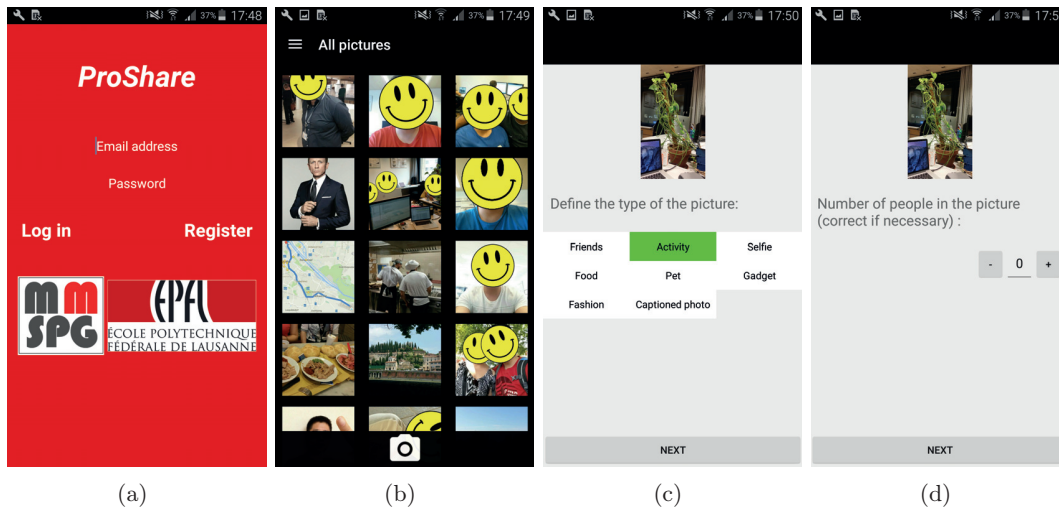
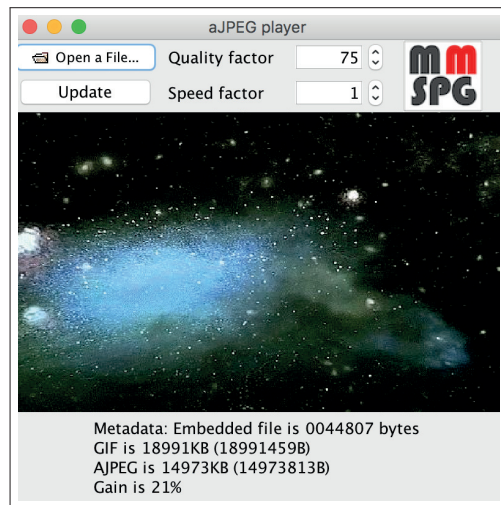
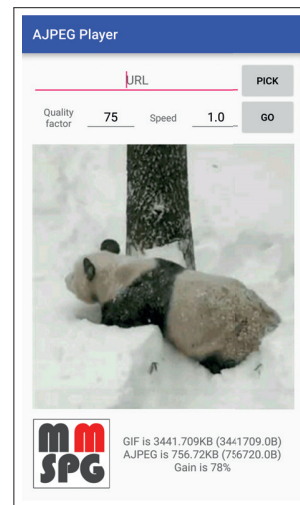


Figure A.6 – Screenshots of ProShare S Android application used for user study in Chapter 7: (a) login page, (b) main page showing all photos, (c)-(f) image semantics annotation, (g) contextual sharing decision questionnaire, (h) protect and upload image.




(a) Desktop app



(b) Mobile Android app

Figure A.7 – Screenshots of two prototype applications for aJPEG conversion and playback in Chapter 8.

**Image:**



**Questions:**

1. Your evoked emotion after seeing this picture is: (select one score from 1 - 9, where 9 corresponds to very positive emotion, 1 to very negative emotion, while others to emotions in between.)

9: very positive ▾

---

2. Confronted with the picture, you are feeling: (select one score from 1 - 9, where 9 corresponds to very excited or stimulated emotion, 1 to calm or relaxed emotion, while others to emotions in between.)

9: excited/stimulated ▾

---

3. Which one(s) of the following keywords best describe your emotion after seeing this picture? (Choose one or more that are suitable)

Joy    Sadness    Fear    Disgust  
 Anger    Surprise    Neutral

---

4. What kind of information of the image influences your evoked emotion the most?

Human facial expression, post or gesture  
 Image color, contrast, saturation, etc.  
 Image background (scene, landmark, etc.)  
 Objects in image (gadgets, clothes, animals, etc.)  
 Texts in image  
 Emoji sticker  
 Halo effect

Figure A.8 – Screenshot of a questionnaire on Microworkers for emotion evaluation of image manipulation in Chapter 9.

Appendix A. Screenshots and Supplied Images

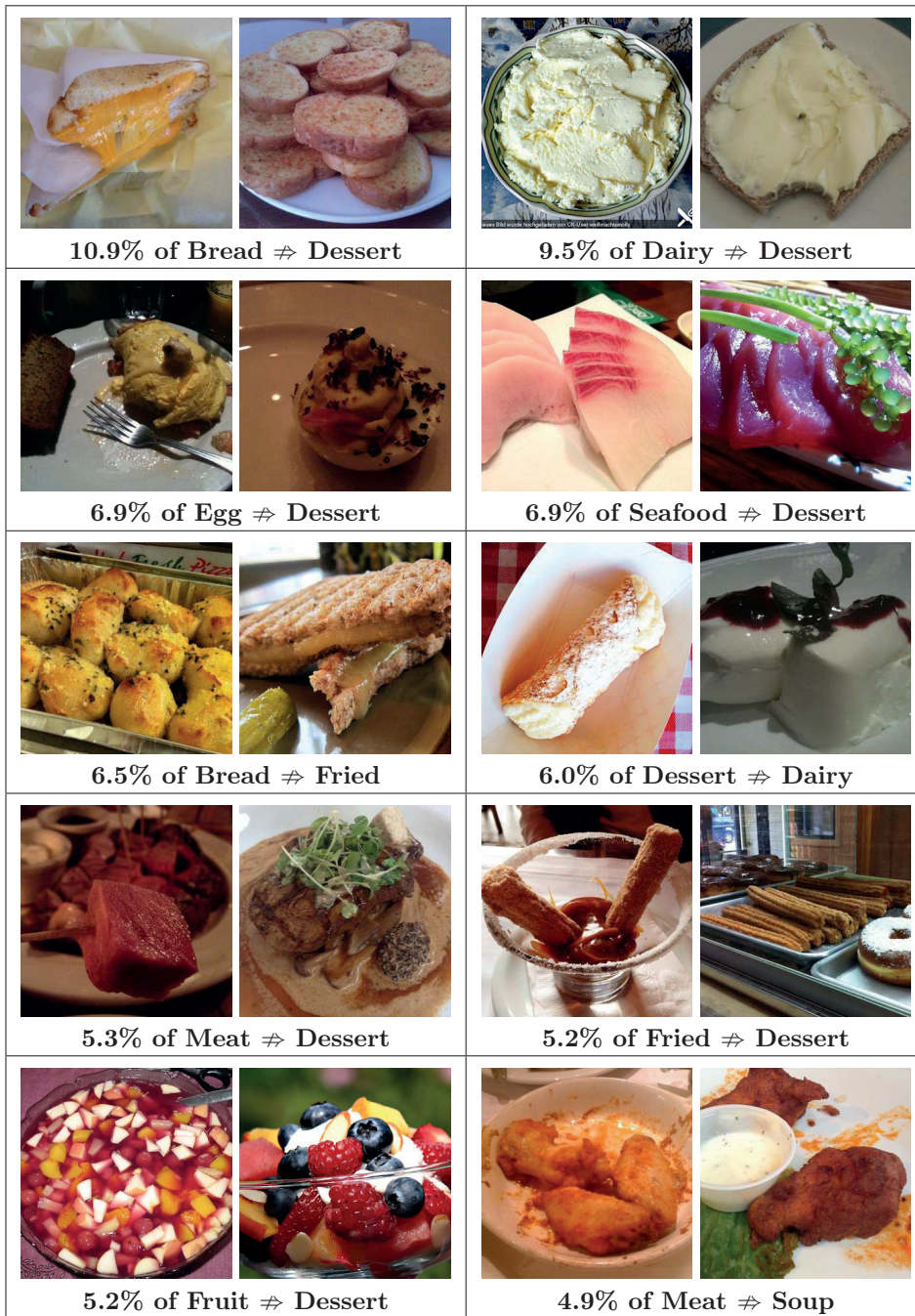


Figure A.9 – Top 10 class pairs that are misclassified in food image categorization experiment in Chapter 10. The percentage refers to the proportion of images in evaluation set for a particular category. The symbol ⇒ stands for “incorrectly classified as”.





# Bibliography

- [1] G. Orwell, *Nineteen Eighty-Four*. Secker & Warburg, 1949.
- [2] J. B. Bayer, N. B. Ellison, S. Y. Schoenebeck, and E. B. Falk, “Sharing the small moments: ephemeral social interaction on Snapchat,” *Information, Communication & Society*, vol. 19, no. 7, pp. 956–977, 2016.
- [3] J. Schiff, M. Meingast, D. K. Mulligan, S. Sastry, and K. Y. Goldberg, “Respectful cameras: Detecting visual markers in real-time to address privacy concerns,” in *IROS*, pp. 971–978, IEEE, 2007.
- [4] D. Chen, Y. Chang, R. Yan, and J. Yang, *Protecting Personal Identification in Video*, pp. 115–128. London: Springer London, 2009.
- [5] J. Wickramasuriya, M. Datt, S. Mehrotra, and N. Venkatasubramanian, “Privacy protecting data collection in media spaces,” in *Proceedings of the 12th Annual ACM International Conference on Multimedia*, (NY, USA), pp. 48–55, 2004.
- [6] S.-C. Cheung, M. Venkatesh, J. Paruchuri, J. Zhao, and T. Nguyen, “Protecting and managing privacy information in video surveillance systems,” in *Protecting Privacy in Video Surveillance*, pp. 11–33, Springer, 2009.
- [7] F. Dufaux and T. Ebrahimi, “Video surveillance using JPEG 2000,” in *Proc. SPIE*, vol. 5558, pp. 268–275, 2004.
- [8] I. Martínez-ponte, X. Desurmont, J. Meessen, and J. François Delaigle, “Robust human face hiding ensuring privacy,” in *in Proc. of International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, 2005.
- [9] T. E. Boult, “PICO: Privacy through Invertible Cryptographic Obscuration,” in *Computer Vision for Interactive and Intelligent Environment (CVIIE’05)*, pp. 27–38, Nov 2005.
- [10] P. Carrillo, H. Kalva, and S. Magliveras, “Compression independent reversible encryption for privacy in video surveillance,” *EURASIP J. Inf. Secur.*, pp. 5:1–5:13, Jan. 2009.
- [11] T. Winkler and B. Rinner, “TrustCAM: Security and Privacy-Protection for an Embedded Smart Camera Based on Trusted Computing,” in *Proceedings of the 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS ’10*, (Washington, DC, USA), pp. 593–600, IEEE Computer Society, 2010.
- [12] A. Pande and J. Zambreno, *Securing Multimedia Content Using Joint Compression and Encryption*, pp. 23–30. London: Springer London, 2013.

## Bibliography

---

- [13] X. Niu, C. Zhou, J. Ding, and B. Yang, "JPEG encryption with file size preservation," in *Intelligent Information Hiding and Multimedia Signal Processing, 2008. IHHMSP'08 International Conference on*, pp. 308–311, IEEE, 2008.
- [14] A. Unterweger and A. Uhl, "Length-preserving Bit-stream-based JPEG Encryption," in *Proceedings of the on Multimedia and security*, pp. 85–90, ACM, 2012.
- [15] C. V. Wright, W.-c. Feng, and F. Liu, "Thumbnail-preserving encryption for JPEG," in *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*, pp. 141–146, ACM, 2015.
- [16] Z. Qian, X. Zhang, and Y. Ren, "JPEG Encryption for Image Rescaling in the Encrypted Domain," *J. Vis. Commun. Image Represent.*, vol. 26, pp. 9–13, Jan. 2015.
- [17] K. He, C. Bidan, and G. Le Guelvouit, "Robust and Secure Image Encryption Schemes During JPEG Compression Process," in *2016 IS&T International Symposium on Electronic Imaging (EI 2016)*, (San Francisco, California, United States), Feb. 2016.
- [18] W. Sun, J. Zhou, R. Lyu, and S. Zhu, "Processing-Aware Privacy-Preserving Photo Sharing over Online Social Networks," in *Proceedings of the 2016 ACM on Multimedia Conference, MM '16*, (New York, NY, USA), pp. 581–585, ACM, 2016.
- [19] J. He, B. Liu, D. Kong, X. Bao, N. Wang, H. Jin, and G. Kesidis, "PUPPIES: Transformation-Supported Personalized Privacy Preserving Partial Image Sharing," in *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 359–370, June 2016.
- [20] A. C. Squicciarini, S. Sundareswaran, D. Lin, and J. Wede, "A3P: Adaptive Policy Prediction for Shared Images over Popular Content Sharing Sites," in *Proceedings of the 22nd ACM Conference on Hypertext and Hypermedia, HT '11*, (New York, NY, USA), pp. 261–270, ACM, 2011.
- [21] A. C. Squicciarini, D. Lin, S. Sundareswaran, and J. Wede, "Privacy policy inference of user-uploaded images on content sharing sites," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, pp. 193–206, Jan 2015.
- [22] L. A. Cutillo, R. Molva, and M. Önen, "Privacy preserving picture sharing: Enforcing usage control in distributed on-line social networks," in *5th ACM Workshop on Social Network Systems*, (Bern, Switzerland), April 2012.
- [23] P. Klemperer, Y. Liang, M. Mazurek, M. Sleeper, B. Ur, L. Bauer, L. F. Cranor, N. Gupta, and M. Reiter, "Tag, you can see it!: Using tags for access control in photo sharing," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12*, (New York, NY, USA), pp. 377–386, ACM, 2012.
- [24] C. Lee, W. Wang, and Y. Guo, "A Fine-Grained Multiparty Access Control Model for Photo Sharing in OSNs," in *2016 IEEE First International Conference on Data Science in Cyberspace (DSC)*, pp. 440–445, June 2016.
- [25] P. Ilia, I. Polakis, E. Athanasopoulos, F. Maggi, and S. Ioannidis, "Face/Off: Preventing Privacy Leakage From Photos in Social Networks," in *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security, CCS '15*, (New York, NY, USA), pp. 781–792, ACM, 2015.

- 
- [26] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, "Persona: An online social network with user-defined privacy," *SIGCOMM Comput. Commun. Rev.*, vol. 39, pp. 135–146, Aug. 2009.
- [27] S. Jahid, P. Mittal, and N. Borisov, "EASiER: Encryption-based access control in social networks with efficient revocation," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, ASIACCS '11*, (New York, NY, USA), pp. 411–415, ACM, 2011.
- [28] S. Guha, K. Tang, and P. Francis, "NOYB: Privacy in online social networks," in *Proceedings of the first workshop on Online social networks*, pp. 49–54, ACM, 2008.
- [29] M. M. Lucas and N. Borisov, "Flybynight: mitigating the privacy risks of social networking," in *Proceedings of the 7th ACM workshop on Privacy in the electronic society*, pp. 1–8, ACM, 2008.
- [30] A. Tootoonchian, K. K. Gollu, S. Saroiu, Y. Ganjali, and A. Wolman, "Lockr: Social Access Control for Web 2.0," in *Proceedings of the First ACM SIGCOMM Workshop on Online Social Networks (WOSN)*, August 2008.
- [31] A. Poller, M. Steinebach, and H. Liu, "Robust image obfuscation for privacy protection in Web 2.0 applications," in *Proc. SPIE*, vol. 8303, pp. 830304–830304–15, 2012.
- [32] M.-R. Ra, R. Govindan, and A. Ortega, "P3: Toward privacy-preserving photo sharing," in *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation*, (Berkeley, CA), pp. 515–528, USENIX, 2013.
- [33] M. Tierney, I. Spiro, C. Bregler, and L. Subramanian, *Cryptagram: Photo privacy for online social media*, pp. 75–87. Association for Computing Machinery, 2013.
- [34] L. Zhang, T. Jung, C. Liu, X. Ding, X.-Y. Li, and Y. Liu, "POP: Privacy-preserving outsourced photo sharing and searching for mobile devices," in *Distributed Computing Systems (ICDCS), 2015 IEEE 35th International Conference on*, pp. 308–317, IEEE, 2015.
- [35] K. He, C. Bidan, and G. Le Guelvouit, "Privacy protection for JPEG content on image-sharing platforms," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, IH&#38;MMSec '16*, (New York, NY, USA), pp. 185–186, ACM, 2016.
- [36] S. Ahern, D. Eckles, N. Good, S. King, M. Naaman, and R. Nair, "Over-exposed?: Privacy patterns and considerations in online and mobile photo sharing," in *CHI*, pp. 357–366, ACM, 2007.
- [37] A. Besmer and H. Richter Lipford, "Moving beyond untagging: Photo privacy in a tagged world," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10*, (New York, NY, USA), pp. 1563–1572, ACM, 2010.
- [38] G. Friedland and R. Sommer, "Cybercasing the Joint: On the Privacy Implications of Geo-tagging," in *Proceedings of the 5th USENIX Conference on Hot Topics in Security, HotSec'10*, (Berkeley, CA, USA), pp. 1–8, USENIX Association, 2010.
- [39] J. a. P. Pesce, D. L. Casas, G. Rauber, and V. Almeida, "Privacy Attacks in Social Media Using Photo Tagging Networks: A Case Study with Facebook," in *Proceedings of the 1st Workshop on Privacy and Security in Online Social Media, PSOSM '12*, (New York, NY, USA), pp. 4:1–4:8, 2012.

## Bibliography

---

- [40] S. Zerr, S. Siersdorfer, J. Hare, and E. Demidova, "Privacy-aware image classification and search," in *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, (New York, NY, USA), pp. 35–44, ACM, 2012.
- [41] L. Tran, D. Kong, H. Jin, and J. Liu, "Privacy-CNH: A framework to detect photo privacy with convolutional neural network using hierarchical features," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [42] A. Tonge and C. Caragea, "Image privacy prediction using deep features," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [43] J. Yu, B. Zhang, Z. Kuang, D. Lin, and J. Fan, "iPrivacy: Image Privacy Protection by Identifying Sensitive Objects via Deep Multi-Task Learning," *IEEE Transactions on Information Forensics and Security*, vol. 12, pp. 1005–1016, May 2017.
- [44] S. J. Oh, R. Benenson, M. Fritz, and B. Schiele, *Faceless Person Recognition: Privacy Implications in Social Media*, pp. 19–35. Cham: Springer International Publishing, 2016.
- [45] R. McPherson, R. Shokri, and V. Shmatikov, "Defeating Image Obfuscation with Deep Learning," *arXiv preprint arXiv:1609.00408*, 2016.
- [46] I. Smith, S. Consolvo, A. Lamarca, J. Hightower, J. Scott, T. Sohn, J. Hughes, G. Iachello, and G. D. Abowd, *Social Disclosure of Place: From Location Technology to Communication Practices*, pp. 134–151. 2005.
- [47] E. Toch, J. Cranshaw, P. H. Drielsma, J. Y. Tsai, P. G. Kelley, J. Springfield, L. Cranor, J. Hong, and N. Sadeh, "Empirical Models of Privacy in Location Sharing," in *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, (NY, USA), pp. 129–138, 2010.
- [48] D. Anthony, T. Henderson, and D. Kotz, "Privacy in Location-Aware Computing Environments," *IEEE Pervasive Computing*, vol. 6, pp. 64–72, Oct. 2007.
- [49] C. Mancini, K. Thomas, Y. Rogers, B. A. Price, L. Jedrzejczyk, A. K. Bandara, A. N. Joinson, and B. Nuseibeh, "From spaces to places: emerging contexts in mobile privacy," in *Proceedings of the 11th international conference on Ubiquitous computing*, pp. 1–10, ACM, 2009.
- [50] N. Sadeh, J. Hong, L. Cranor, I. Fette, P. Kelley, M. Prabaker, and J. Rao, "Understanding and capturing people's privacy policies in a mobile social networking application," *Personal and Ubiquitous Computing*, vol. 13, no. 6, pp. 401–412, 2009.
- [51] M. Benisch, P. G. Kelley, N. Sadeh, and L. F. Cranor, "Capturing location-privacy preferences: quantifying accuracy and user-burden tradeoffs," *Personal and Ubiquitous Computing*, vol. 15, no. 7, pp. 679–694, 2011.
- [52] J. Wiese, P. G. Kelley, L. F. Cranor, L. Dabbish, J. I. Hong, and J. Zimmerman, "Are you close with me? are you nearby?: investigating social groups, closeness, and willingness to share," in *Proceedings of the 13th international conference on Ubiquitous computing*, pp. 197–206, ACM, 2011.
- [53] L. Fang and K. LeFevre, "Privacy wizards for social networking sites," in *Proceedings of the 19th international conference on World wide web*, pp. 351–360, ACM, 2010.

- 
- [54] G. Bigwood, F. B. Abdesslem, and T. Henderson, “Predicting Location-Sharing Privacy Preferences in Social Network Applications,” in *In Proc. of AwareCast*, 2012.
- [55] J. Xie, B. P. Knijnenburg, and H. Jin, “Location Sharing Privacy Preference: Analysis and Personalized Recommendation,” in *Proceedings of the 19th International Conference on Intelligent User Interfaces*, pp. 189–198, 2014.
- [56] H. Harkous, R. Rahman, and K. Aberer, “C3P: Context-Aware Crowdsourced Cloud Privacy,” in *Privacy Enhancing Technologies, Pets 2014*, vol. 8555, pp. 102–122, Springer-Verlag Berlin, 2014.
- [57] I. Bilogrevic, K. Huguenin, B. Agir, M. Jadhwal, M. Gazaki, and J.-P. Hubaux, “A machine-learning based approach to privacy-aware information-sharing in mobile social networks,” *Pervasive and Mobile Computing*, vol. 25, pp. 125 – 142, 2016.
- [58] F. Dufaux and T. Ebrahimi, “Toward a Secure JPEG,” in *Proc. SPIE*, vol. 6312, 2006.
- [59] ITU-T, “JPEG Standard, JPEG ISO/IEC 10918-1,” tech. rep., 1993.
- [60] Ecma International, “JPEG File Interchange Format (JFIF),” tech. rep., 2009.
- [61] “Exchangeable image file format for digital still cameras: Exif Version 2.2,” 2002. Standard of Japan Electronics and Information Technology Industries Association.
- [62] “JPSEC final draft international standard.” ISO/IEC JTC1/SC29/WG1/N3820, Nov 2005.
- [63] J. Apostolopoulos, S. Wee, F. Dufaux, T. Ebrahimi, Q. Sun, and Z. Zhang, “The emerging JPEG-2000 security (JPSEC) standard,” in *2006 IEEE International Symposium on Circuits and Systems*, pp. 4 pp.–3885, May 2006.
- [64] F. Dufaux and T. Ebrahimi, “Scrambling for privacy protection in video surveillance systems,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, pp. 1168–1174, Aug 2008.
- [65] A. G. Weber, “The USC-SIPI Image Database,” tech. rep., University of Southern California, Signal and Image Processing Institute, Department of Electrical Engineering, Los Angeles, CA 90089-2564 USA, 3740 McClintock Ave, Oct. 1997.
- [66] N. Zhang, M. Paluri, Y. Taigman, R. Fergus, and L. Bourdev, “Beyond frontal faces: Improving person recognition using multiple cues,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4804–4813, June 2015.
- [67] H. Jegou, M. Douze, and C. Schmid, “Hamming embedding and weak geometric consistency for large scale image search,” in *Proceedings of the 10th European Conference on Computer Vision: Part I, ECCV '08, (Berlin, Heidelberg)*, pp. 304–317, Springer-Verlag, 2008.
- [68] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, pp. I–511–I–518 vol.1, 2001.
- [69] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces,” in *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pp. 586–591, IEEE, 1991.
- [70] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

## Bibliography

---

- [71] T. Ahonen, A. Hadid, and M. Pietikäinen, “Face recognition with local binary patterns,” in *European conference on computer vision*, pp. 469–481, Springer, 2004.
- [72] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, “The FERET evaluation methodology for face-recognition algorithms,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.
- [73] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, April 2004.
- [74] J. Daemen and V. Rijmen, *The Design of Rijndael*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2002.
- [75] J. W. Moon, J. S. Lee, and N. I. Cho, “A requantization algorithm for the transcoding of JPEG images,” *Signal Processing: Image Communication*, vol. 21, no. 1, pp. 13–21, 2006.
- [76] M. M. Bradley and P. J. Lang, “Measuring emotion: the Self-Assessment Manikin and the Semantic Differential,” *Journal of behavior therapy and experimental psychiatry*, vol. 25, no. 1, pp. 49–59, 1994.
- [77] K. Pearson, “Note on regression and inheritance in the case of two parents,” *Proceedings of the Royal Society of London*, vol. 58, pp. 240–242, 1895.
- [78] J. Daemen and V. Rijmen, *The Design of Rijndael*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2002.
- [79] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2414–2423, June 2016.
- [80] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Commun. ACM*, vol. 21, pp. 120–126, Feb. 1978.
- [81] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS ’06*, (New York, NY, USA), pp. 89–98, ACM, 2006.
- [82] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” in *Proceedings of the 2007 IEEE Symposium on Security and Privacy, SP ’07*, (Washington, DC, USA), pp. 321–334, IEEE Computer Society, 2007.
- [83] B. Waters, *Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization*, pp. 53–70. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.
- [84] B. Kaliski, “RFC 2898 - PKCS #5: Password-Based Cryptography Specification Version 2.0,” tech. rep., IETF, Sept. 2000.
- [85] Z. Xu and K. Martin, “Dynamic user revocation and key refreshing for attribute-based encryption in cloud storage,” in *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 844–849, June 2012.
- [86] Y. Hu and L. M. Y. S. Kambhampati, “What we Instagram: A first analysis of instagram photo content and user types,” in *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*, pp. 595–598, The AAAI Press, 2014.

- 
- [87] D. Castro, S. Hickson, V. Bettadapura, E. Thomaz, G. Abowd, H. Christensen, and I. Essa, "Predicting daily activities from egocentric images using deep learning," *ISWC*, 2015.
- [88] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, pp. 10–18, Nov. 2009.
- [89] A. J. Viera, J. M. Garrett, *et al.*, "Understanding interobserver agreement: the kappa statistic," *Fam Med*, vol. 37, no. 5, pp. 360–363, 2005.
- [90] C. Elkan, "The foundations of cost-sensitive learning," in *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'01, (San Francisco, CA, USA), pp. 973–978, Morgan Kaufmann Publishers Inc., 2001.
- [91] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708, June 2014.
- [92] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.
- [93] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference*, 2015.
- [94] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, June 2015.
- [95] S. J. Oh, R. Benenson, M. Fritz, and B. Schiele, "Person recognition in personal photo collections," in *ICCV*, 2015.
- [96] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.
- [97] A. Furnari, G. M. Farinella, and S. Battiato, "Recognizing personal contexts from egocentric images," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015.
- [98] M. Bolanos, M. Dimiccoli, and P. Radeva, "Toward storytelling from visual lifelogging: An overview," *IEEE Transactions on Human-Machine Systems*, vol. 47, pp. 77–90, Feb 2017.
- [99] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in neural information processing systems*, pp. 487–495, 2014.
- [100] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929, 2016.
- [101] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva, "Places: An image database for deep scene understanding," *arXiv preprint arXiv:1610.02055*, 2016.
- [102] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

## Bibliography

---

- [103] J. Ziv and A. Lempel, "Compression of Individual Sequences via Variable-rate Coding," *IEEE Trans. Inf. Theor.*, vol. 24, pp. 530–536, Sept. 2006.
- [104] T. A. Welch, "A technique for high-performance data compression," *Computer*, vol. 17, pp. 8–19, June 1984.
- [105] T. Welch, "High speed data compression and decompression apparatus and method," Dec. 1985. US Patent 4,558,302.
- [106] M. Flickinger, "What's In A GIF - Bit by Byte," 2016. [Online; accessed Mai-2016].
- [107] S. Parmenter, V. Vukicevic, and A. Smith, "APNG Specification." [https://wiki.mozilla.org/APNG\\_Specification](https://wiki.mozilla.org/APNG_Specification), 2015. [Online; accessed Mai-2016].
- [108] G. Roelofs, "Multiple-image Network Graphics." <http://www.libpng.org/pub/mng/>, 2015. [Online; accessed Mai-2016].
- [109] Y. Li, Y. Song, L. Cao, J. R. Tetreault, L. Goldberg, A. Jaimes, and J. Luo, "TGIF: A New Dataset and Benchmark on Animated GIF Description," *CoRR*, vol. abs/1604.02748, 2016.
- [110] F. D. Simone, M. Tagliasacchi, M. Naccari, S. Tubaro, and T. Ebrahimi, "A H.264/AVC video database for the evaluation of quality metrics," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2430–2433, March 2010.
- [111] F. D. Simone, M. Naccari, M. Tagliasacchi, F. Dufaux, S. Tubaro, and T. Ebrahimi, "Subjective assessment of H.264/AVC video sequences transmitted over a noisy channel," in *Quality of Multimedia Experience, 2009. QoMEx 2009. International Workshop on*, pp. 204–209, July 2009.
- [112] D. Joshi, R. Datta, E. Fedorovskaya, Q.-T. Luong, J. Z. Wang, J. Li, and J. Luo, "Aesthetics and emotions in images," *Signal Processing Magazine, IEEE*, vol. 28, no. 5, pp. 94–115, 2011.
- [113] J. Machajdik and A. Hanbury, "Affective image classification using features inspired by psychology and art theory," in *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pp. 83–92, ACM, 2010.
- [114] M. Solli and R. Lenz, "Emotion related structures in large image databases," in *Proceedings of the ACM International Conference on Image and Video Retrieval, CIVR '10*, (New York, NY, USA), pp. 398–405, ACM, 2010.
- [115] X. Wang, J. Jia, J. Yin, and L. Cai, "Interpretable aesthetic features for affective image classification," in *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pp. 3230–3234, Sept 2013.
- [116] S. Zhao, Y. Gao, X. Jiang, H. Yao, T.-S. Chua, and X. Sun, "Exploring principles-of-art features for image emotion recognition," in *Proceedings of the 22nd ACM International Conference on Multimedia*, MM '14, pp. 47–56.
- [117] Y.-Y. Chen, T. Chen, W. H. Hsu, H.-Y. M. Liao, and S.-F. Chang, "Predicting viewer affective comments based on image content in social media," in *ICMR*, pp. 233:233–233:240, 2014.
- [118] B. Jou, S. Bhattacharya, and S.-F. Chang, "Predicting viewer perceived emotions in animated GIFs," in *Proceedings of the 22nd ACM International Conference on Multimedia*, MM '14, (New York, NY, USA), pp. 213–216, ACM, 2014.



- [119] X. Wang, J. Jia, and L. Cai, "Affective image adjustment with a single word," *Vis. Comput.*, vol. 29, pp. 1121–1133, Nov. 2013.
- [120] K.-C. Peng, K. Karlsson, T. Chen, D.-Q. Zhang, and H. Yu, "A framework of changing image emotion using emotion prediction," in *Image Processing (ICIP), 2014 IEEE International Conference on*, Oct 2014.
- [121] J. Jun, L.-C. Ou, B. Oicherman, S.-T. Wei, M. R. Luo, H. Nachilieli, and C. Staelin, "Psychophysical and psychophysiological measurement of image emotion," in *Color and Imaging Conference*, no. 1, pp. 121–127, Society for Imaging Science and Technology, 2010.
- [122] K.-C. Peng, T. Chen, A. Sadovnik, and A. C. Gallagher, "A mixed bag of emotions: Model, predict, and transfer emotion distributions," in *CVPR*, pp. 860–868, IEEE, 2015.
- [123] J. L. Starck, F. Murtagh, E. J. Candes, and D. L. Donoho, "Gray and color image contrast enhancement by the curvelet transform," *IEEE Transactions on Image Processing*, vol. 12, pp. 706–717, June 2003.
- [124] C. Wang, J. Zhang, B. Yang, and L. Zhang, "Sketch2Cartoon: Composing cartoon images by sketching," in *Proc. 19th ACM MM*, pp. 789–790.
- [125] L. Yuan and T. Ebrahimi, "Image Transmorphing with JPEG," in *Image Processing (ICIP), 2015 IEEE International Conference on*, pp. 3956–3960, Sept 2015.
- [126] P. J. Lang, M. M. Bradley, and B. N. Cuthbert, "International affective picture system (IAPS): Affective ratings of pictures and instruction manual," Tech. Rep. A-8, The Center for Research in Psychophysiology, University of Florida, Gainesville, FL, 2008.
- [127] E. S. Dan-Glauser and K. R. Scherer, "The Geneva affective picture database (GAPED): a new 730-picture database focusing on valence and normative significance," *Behavior research methods*, vol. 43, no. 2, pp. 468–477, 2011.
- [128] J. Russell, "A circumplex model of affect," *Journal of personality and social psychology*, vol. 39, no. 6, pp. 1161–1178, 1980.
- [129] P. Ekman, W. V. Friesen, and P. Ellsworth, "CHAPTER XIII - what emotion categories can observers judge from facial behavior?," in *Emotion in the Human Face*, vol. 11 of *Pergamon General Psychology Series*, pp. 57 – 65, Pergamon, 1972.
- [130] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414–2423, 2016.
- [131] D. Hasler and S. E. Suesstrunk, "Measuring colorfulness in natural images," in *Proc. SPIE*, vol. 5007, pp. 87–95, 2003.
- [132] K. Matković, L. Neumann, A. Neumann, T. Psik, and W. Purgathofer, "Global contrast factor - a new approach to image contrast," in *Proceedings of the First Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging*, pp. 159–167, 2005.
- [133] F. Pedregosa, G. Varoquaux, A. Gramfort, and et al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [134] G. M. Farinella, D. Allegra, M. Moltisanti, F. Stanco, and S. Battiato, "Retrieval and classification of food images," *Computers in Biology and Medicine*, vol. 77, pp. 23 – 39, 2016.

## Bibliography

---

- [135] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, (New York, NY, USA), pp. 675–678, ACM, 2014.
- [136] D. G. Lowe, "Object recognition from local scale-invariant features," in *The proceedings of the 7th IEEE international conference on Computer Vision*, vol. 2, pp. 1150–1157, 1999.
- [137] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [138] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 2169–2178, 2006.
- [139] J. Sanchez, F. Perronmin, T. Mensink, and J. Verbeek, "Image Classification with the Fisher Vector: Theory and Practice," *International Journal of Computer Vision*, vol. 105, pp. 222–245, Dec. 2013.
- [140] K. Kitamura, T. Yamasaki, and K. Aizawa, "Food log by analyzing food images," in *Proceedings of the 16th ACM International Conference on Multimedia*, pp. 999–1000, 2008.
- [141] J. Nie, Z. Wei, W. Jia, L. Li, J. D. Fernstrom, R. J. Sclabassi, and M. Sun, "Automatic detection of dining plates for image-based dietary evaluation," in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pp. 4312–4315, Aug 2010.
- [142] G. M. Farinella, D. Allegra, F. Stanco, and S. Battiato, *On the Exploitation of One Class Classification to Distinguish Food Vs Non-Food Images*, pp. 375–383. Cham: Springer International Publishing, 2015.
- [143] X. Qi, R. Xiao, C.-G. Li, Y. Qiao, J. Guo, and X. Tang, "Pairwise rotation invariant co-occurrence local binary pattern," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2199–2213, 2014.
- [144] G. M. Farinella, M. Moltisanti, and S. Battiato, "Classifying food images represented as bag of textons," in *Image Processing (ICIP), 2014 IEEE International Conference on*, pp. 5212–5216, IEEE, 2014.
- [145] G. M. Farinella, D. Allegra, and F. Stanco, "A benchmark dataset to study the representation of food images," in *European Conference on Computer Vision*, pp. 584–599, Springer, 2014.
- [146] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, Nov 1998.
- [147] H. Kagaya, K. Aizawa, and M. Ogawa, "Food detection and recognition using convolutional neural network," in *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, (New York, NY, USA), pp. 1085–1088, ACM, 2014.
- [148] H. Kagaya and K. Aizawa, "Highly accurate food/non-food image classification based on a deep convolutional neural network," in *ICIAP 2015 Workshop: MADiMa*, pp. 350–357, 2015.
- [149] T. Joutou and K. Yanai, "A food image recognition system with multiple kernel learning," in *IEEE ICIP*, pp. 285–288, Nov 2009.
- [150] H. Hoashi, T. Joutou, and K. Yanai, "Image recognition of 85 food categories by feature fusion," in *IEEE International Symposium on Multimedia (ISM)*, pp. 296–301, 2010.

- 
- [151] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang, "PFID: Pittsburgh fast-food image dataset," in *IEEE ICIP*, pp. 289–292, Nov 2009.
- [152] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, "Food recognition using statistics of pairwise local features," in *CVPR*, pp. 2249–2256, June 2010.
- [153] V. Bettadapura, E. Thomaz, A. Parnami, G. D. Abowd, and I. A. Essa, "Leveraging context to support automated food recognition in restaurants," *CoRR*, vol. abs/1510.02078, 2015.
- [154] M. H. Rahmana, M. R. Pickering, D. Kerr, C. J. Boushey, and E. J. Delp, "A new texture feature for improved food recognition accuracy in a mobile phone based dietary assessment system," in *IEEE ICMEW*, pp. 418–423, July 2012.
- [155] Y. He, C. Xu, N. Khanna, C. J. Boushey, and E. J. Delp, "Analysis of food images: Features and classification," in *IEEE ICIP*, pp. 2744–2748, Oct 2014.
- [156] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Transactions on circuits and systems for video technology*, vol. 11, no. 6, pp. 703–715, 2001.
- [157] K.-M. Wong, L.-M. Po, and K.-W. Cheung, "Dominant color structure descriptor for image retrieval," in *IEEE ICIP*, vol. 6, pp. VI–365, IEEE, 2007.
- [158] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2547–2554, 2013.
- [159] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 – mining discriminative components with random forests," in *European Conference on Computer Vision*, 2014.
- [160] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, pp. 1097–1105, Curran Associates, Inc., 2012.
- [161] Y. Kawano and K. Yanai, "Food image recognition with deep convolutional features," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pp. 589–593, ACM, 2014.
- [162] Y. Matsuda, H. Hoashi, and K. Yanai, "Recognition of multiple-food images by detecting candidate regions," in *IEEE ICME*, 2012.
- [163] K. Yanai and Y. Kawano, "Food image recognition using deep convolutional network with pre-training and fine-tuning," in *Multimedia & Expo Workshops (ICMEW), 2015 IEEE International Conference on*, pp. 1–6, IEEE, 2015.
- [164] Y. Kawano and K. Yanai, "Automatic expansion of a food image dataset leveraging existing categories with domain adaptation," in *Proc. of ECCV Workshop on Transferring and Adapting Source Knowledge in Computer Vision (TASK-CV)*, 2014.
- [165] A. Myers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. Murphy, "Im2Calories: towards an automated mobile vision food diary," in *ICCV*, 2015.
- [166] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Computer Vision and Pattern Recognition (CVPR)*, 2015.

## Bibliography

---

- [167] H. Wang and B. Raj, “A survey: Time travel in deep learning space: An introduction to deep learning models and how deep learning models evolved from the initial ideas,” *CoRR*, vol. abs/1510.04781, 2015.
- [168] M. Lin, Q. Chen, and S. Yan, “Network in network,” *CoRR*, vol. abs/1312.4400, 2013.
- [169] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “OverFeat: Integrated recognition, localization and detection using convolutional networks,” *CoRR*, vol. abs/1312.6229, 2013.
- [170] Z. Zhong, L. Jin, and Z. Xie, “High performance offline handwritten chinese character recognition using googlenet and directional feature maps,” in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pp. 846–850, Aug 2015.
- [171] L. Torrey and J. Shavlik, “Transfer learning,” in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques 1*, 2009.
- [172] L. Y. Pratt, “Discriminability-based transfer between neural networks,” in *Advances in Neural Information Processing Systems 5, [NIPS Conference]*, (San Francisco, CA, USA), pp. 204–211, Morgan Kaufmann Publishers Inc., 1993.
- [173] S. Thrun and L. Pratt, *Learning to learn*. Springer Science & Business Media, 2012.
- [174] R. Caruana, “Multitask learning,” *Mach. Learn.*, vol. 28, pp. 41–75, July 1997.
- [175] R. Caruana, “Multitask learning,” in *Learning to learn*, pp. 95–133, Springer, 1998.
- [176] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255, IEEE, 2009.
- [177] A. Karpathy, “Stanford University CS231n: Convolutional Neural Networks for Visual Recognition,”
- [178] L. Fei-Fei, R. Fergus, and P. Perona, “Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories,” in *Computer Vision and Pattern Recognition Workshop*, pp. 178–178, June 2004.
- [179] G. Griffin, A. Holub, and P. Perona, “Caltech-256 object category dataset,” Tech. Rep. 7694, California Institute of Technology, 2007.
- [180] A. Gallagher and T. Chen, “Understanding images of groups of people,” in *CVPR*, 2009.
- [181] M. Nestle, *Food politics: How the food industry influences nutrition and health*, vol. 3. Univ of California Press, 2013.
- [182] D. M. Powers, “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation,” 2011.
- [183] J. Cohen, “A Coefficient of Agreement for Nominal Scales,” *Educational and Psychological Measurement*, vol. 20, no. 1, p. 37, 1960.
- [184] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [185] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *arXiv preprint arXiv:1512.03385*, 2015.

## Lin Yuan

Avenue de Tivoli 64  
1007 Lausanne  
Switzerland  
Phone: +41 76 783 5836  
Email: [lin.yuan@epfl.ch](mailto:lin.yuan@epfl.ch)

Date of birth: June 6, 1989  
Nationality: Chinese  
Marital status: Single



---

### SUMMARY

- Strong background in Signal/Image Processing, Multimedia Security & Privacy, Computer Vision, Machine Learning
- Experienced skills in Python programming, Mobile (iOS and Android) and Web development
- Fast learner, capable of strong teamwork and management skills.

---

### EDUCATION

- Mar. 2014 - Present    **Ph.D. in Electrical Engineering**  
**Swiss Federal Institute of Technology in Lausanne (EPFL)**, Switzerland  
- Thesis: Privacy-Friendly Photo Sharing and Relevant Applications Beyond  
- Advisor: **Prof. Touradj Ebrahimi**
- Sep. 2011 - Aug 2013    **M.Sc. in Electrical and Electronics Engineering**  
**Swiss Federal Institute of Technology in Lausanne (EPFL)**, Switzerland  
- Major in Information Technologies (GPA: 5.19/6)  
- Thesis: Simulating Robustness of Audio Watermarks against Acoustic Path
- Sep. 2007 - July 2011    **B.Sc. in Electronic Science and Technology**  
**University of Electronic Science and Technology of China (UESTC)**,  
Chengdu, China  
- Major in Optoelectronic Engineering (GPA: 3.64/4, Ranking: 15/160)  
- Awarded three years UESTC People's Scholarships & the Outstanding Graduate of Sichuan

---

### WORK EXPERIENCE

- Mar. 2014 - present    **Multimedia Signal Processing Group (MMSPG), EPFL**, Lausanne, Switzerland  
**Research & Teaching Assistant**  
- Supervision of 13 Bachelor/Master students' projects and internships  
- Responsible TA of two master courses: Image and Video Processing, Media Security  
- Active contributions to JPEG Privacy & Security Standardization
- July 2012 - Aug. 2013    **Content Processing Laboratory, Technicolor R&I**, Hannover, Germany  
**Intern**  
- Research, development and evaluation of audio watermarking benchmark system

---

### SKILLS

- Programming:        Python, Matlab, C, Objective-C, Java, Web development (HTML, CSS, JavaScript, PHP & SQL), Mobile development (iOS & Android)
- Specific Software:    Weka, Caffe, Scikit-learn, OpenCV, SciPy, FFmpeg, Blender, L<sup>A</sup>T<sub>E</sub>X, Microsoft Office, Git
- Operating Systems:    Linux, Mac OS, Windows

---

### LANGUAGES

Chinese (mother tongue), English (fluent), French (beginner)

## PUBLICATIONS

---

**Lin Yuan** and Touradj Ebrahimi. “Image Privacy Protection with Secure JPEG Transmorphing”. *Journal of IET Signal Processing* (Submitted and under peer review).

**Lin Yuan**, Joël Theytaz and Touradj Ebrahimi. “Context-Dependent Privacy-Aware Photo Sharing based on Machine Learning”. 32nd International Conference on ICT Systems Security and Privacy Protection (IFIP SEC 2017), Rome, Italy.

**Lin Yuan** and Touradj Ebrahimi. “Evaluation and Prediction of Evoked Emotions Induced by Image Manipulations”. *Human Vision and Electronic Imaging (HVEI) 2017*, San Francisco, 2017.

Joël Theytaz, **Lin Yuan**, David McNally, and Touradj Ebrahimi. “Towards an animated JPEG”. *Proc. SPIE 9971, Applications of Digital Image Processing XXXIX, 99711X* (September 28, 2016); doi:10.1117/12.2240283.

Ashutosh Singla, **Lin Yuan**, and Touradj Ebrahimi. “Food/Non-food Image Classification and Food Categorization using Pre-Trained GoogLeNet Model”. In *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management (MADiMa '16)*. DOI: <https://doi.org/10.1145/2986035.2986039>.

**Lin Yuan**, David McNally, Alptekin Küpçü, and Touradj Ebrahimi. “Privacy-Preserving Photo Sharing based on a Public Key Infrastructure”. *Proc. SPIE 9599, Applications of Digital Image Processing XXXVIII, 95991I* (September 22, 2015); doi:10.1117/12.2190458.

**Lin Yuan** and Touradj Ebrahimi. “Image Transmorphing with JPEG”. 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, 2015, pp. 3956-3960. doi: 10.1109/ICIP.2015.7351547.

Martin Rerabek, **Lin Yuan**, Leonard Authier, and Touradj Ebrahimi. [ISO/IEC JTC 1/SC 29/WG1 contribution] “EPFL Light-Field Image Dataset”, 2015.

**Lin Yuan**, Pavel Korshunov, and Touradj Ebrahimi. “Secure JPEG Scrambling Enabling Privacy in Photo Sharing”. 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), Ljubljana, 2015, pp. 1-6. doi: 10.1109/FG.2015.7285022.

**Lin Yuan**, Pavel Korshunov, and Touradj Ebrahimi. “Privacy-Preserving Photo Sharing based on a Secure JPEG”. 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPs), Hong Kong, 2015, pp. 185-190. doi: 10.1109/INFOCOMW.2015.7179382.

Martin Rerabek, **Lin Yuan**, Lukas Krasula, Pavel Korshunov, Karel Fliegel, and Touradj Ebrahimi. “Evaluation of Privacy in High Dynamic Range Video Sequences”. *Proc. SPIE 9217, Applications of Digital Image Processing XXXVII, 92170E* (September 23, 2014); doi:10.1117/12.2065559.

Xiaobo Gu, Zhenming Peng, Zhiwei Chen, **Lin Yuan**, Bingquan Huang. “Image Fusion using Lifting Wavelet Transform with Human Visual Features”. *Proc. SPIE 7850, Optoelectronic Imaging and Multimedia Technology, 78502L* (November 10, 2010); doi:10.1117/12.871546.

## PATENT

---

Touradj Ebrahimi and **Lin Yuan**. “Media content processing method”. US20170034523 A1. February 2, 2017. <https://www.google.com/patents/US20170034523>.

