EPFL

Thèse n° 8979

Human-Centered Scene Understanding via Crowd Counting

Présentée le 26 novembre 2021

Faculté informatique et communications
Laboratoire de vision par ordinateur
Programme doctoral en informatique et communications

pour l'obtention du grade de Docteur ès Sciences

par

Weizhe LIU

Acceptée sur proposition du jury

Dr M. Rajman, président du jury
Prof. P. Fua, directeur de thèse
Dr I. Laptev, rapporteur
Prof. D. Samaras, rapporteur
Prof. S. Süsstrunk, rapporteuse

École polytechnique fédérale de Lausanne

2021

To my parents…

Acknowledgements

This thesis would not be possible without the help and support of many people. I take this opportunity to express my gratitude to all of them.

First and foremost, I would like to thank my advisor Prof. Pascal Fua. I am immensely grateful to Pascal for offering me the opportunity to pursue my studies in his laboratory. I remember so many invaluable discussions we had during the past four years and I am always impressed by his professionalism, honesty and erudition. Pascal not only teaches me how to do research but also shows me the necessary quality of a life-long researcher – constant thirst for new knowledge.

I would like to also thank Dr. Mathieu Salzmann for the helpful discussion and valuable feedback during my PhD study. Our discussions helped me realize the importance of simplicity and preciseness in writing scientific paper.

I am very thankful to my thesis jury members, Dr. Martin Rajman, Dr. Ivan Laptev, Prof. Dimitris Samaras and Prof. Sabine Süsstrunk for kindly accepting to evaluate this work and providing valuable feedback.

Many thanks to Dr. Christian Leistner for his great guidance during my internship at Amazon which helped me realize the difference between academia and industry. It also motivates me to address computer vision problems in real-world scenarios where the data annotation is not always available. I would like to also thank Dr. David Ferstl, Dr. Samuel Schulter and Dr. Lukas Zebedin for helpful discussions during this internship.

I am very grateful to Dr. Bugra Tekin. We met each other when we were both pursuing our studies at Computer Vision Lab and he became my mentor during my internship at Microsoft later. Bugra not only provided me insightful guidance on research project but also gave me many valuable suggestions in life, which made the internship a productive and joyful experience for me. I want to also thank Dr. Huseyin Coskun, Dr. Vibhav Vineet and Taein Kwon for helpful discussions during this internship.

I am very fortune to be a member of Computer Vision Lab where I met many wonderful people and I would like to thank all of them. To our secretary Ariane Staudenmann, for her help to solve many administrative problems that come along the way. To my officemates Shuxuan, Isinsu and Chen, for the relaxed daily chats. To Krzysztof and Nikita, for their assistance and effort in preparing our papers. To Yinlin and Zheng, for their helpful discussions in various research topics

and valuable suggestions in writing scientific paper. To Kaicheng, Sena and Jan, for organizing meals and games that bring so much fun to my life. To Helge, for proofreading my master thesis. To Wei, Krishna, Vidit and Udaranga, for many discussions we had in the past years. To Edoardo, Benoît, Zhen, Jiancheng, Mengshi, Doruk, Mateusz, Leonardo, Victor, Joachim, Martin, Amaury, Okan, Semih, Sina, Shaifali, Kwang, Pierre, Timur, Agata, Eduard, Pablo, Ksenia and many others for pleasant interactions. Thanks to them, this PhD study has been a great experience for me.

I would like to thank many people I met during my master thesis in Los Angeles. To Prof. Stefano Soatto, for supervising my master thesis, without his recommendation I would not have the chance to pursuit my PhD studies. To Xiaohan, for helpful discussions and his generousness to share his GPU with me. To Tong, for helping me adjust to the life in Los Angeles. To Jingming, Peng, Xinzhu, Nikos, Isaac and Kareem for all the fun memories.

I owe my deepest gratitude to all my friends I met in Lausanne who bring countless joy to my life. I would like to thank Junli for his encouragement in course study. I want to thank Jin for her suggestion of my master internship, which was the first computer vision project I ever did and it motivated me to further pursuit my PhD study in this area. I am very thankful to my EDIC friends, Su, Cong, Tao, Chen and Ruofan for daily chats. In particular, I would like to thank my mathematician friends, to Shengquan and Zhiwen for their encouragement and company. To Haoqing, for many amusing chats and free fitness coaching. To my buddy Hao, we came to EPFL by the same airplane in 2014 and I would like to thank his kindness and invaluable support through my master and PhD studies.

Most importantly, I want to acknowledge the unconditional support of my parents, to whom the thesis is dedicated. Without their endless love, I would not have been able to move forward during my PhD study.

Lausanne, October 2021 Weizhe Liu

Abstract

Human-centered scene understanding is the process of perceiving and analysing a dynamic scene observed through a network of sensors with emphasis on human-related activities. It includes the visual perception of human-related activities from either single image or video sequence. Scene understanding with focus of human-related activities is becoming increasingly popular which results in the demand of algorithms that can efficiently model crowd activity in different real-world scenarios.

In this thesis, we exploit human-centered scene understanding through crowd counting. Counting people is a challenging task due to perspective distortion and occlusion. We tackle these problems by developing algorithms to leverage a variety of data modalities including single image, video sequence and scene perspective map.

First, we introduce an end-to-end trainable deep architecture for crowd counting that combines features obtained using multiple receptive field sizes and learns the importance of each such feature at each image location. In other words, our approach adaptively encodes the scale of the contextual information required to accurately predict crowd density. This yields an algorithm that outperforms previous crowd counting methods, especially when perspective effects are strong.

Second, we explicitly model the scale changes and reason in terms of people per square-meter. We show that feeding the perspective model to the network allows us to enforce global scale consistency and that this model can be obtained on the fly from the drone sensors. In addition, it also enables us to enforce physically-inspired temporal consistency constraints that do not have to be learned. This yields an algorithm that outperforms previous methods in inferring crowd density from a moving drone camera especially when perspective effects are strong.

Third, for video sequence, we advocate estimating people flows across image locations between consecutive images and inferring the people densities from these flows instead of directly regressing them. This enables us to impose much stronger constraints encoding the conservation of the number of people. As a result, it significantly boosts performance without requiring a more complex architecture. Furthermore, it allows us to exploit the correlation between people flow and optical flow to further improve the results. We also show that leveraging people conservation constraints in both a spatial and temporal manner makes it possible to train a deep crowd counting model in an active learning setting with much fewer annotations. This significantly reduces the annotation cost while still leading to similar performance to the full supervision case.

Keywords: scene understanding, crowd counting, deep neural networks

Résumé

La compréhension de scène centrée sur l'humain est le processus de perception et d'analyse d'une scène dynamique observée à travers un réseau de capteurs en mettant l'accent sur les activités humaines. Il comprend la perception visuelle des activités humaines à partir d'une seule image ou d'une séquence vidéo. La compréhension de scènes centrées sur les activités humaines devient de plus en plus populaire, ce qui nécessite le développement d'algorithmes capables de modéliser efficacement l'activité des foules dans différents scénarios.

Dans cette thèse, nous abordons la compréhension de scène centrée sur l'humain à travers le comptage de foule. Compter les personnes est une tâche difficile en raison de la distorsion et de l'occlusion de la perspective. Nous abordons ces problèmes en développant des algorithmes pour tirer parti d'une variété de modalités de données, notamment une image unique, une séquence vidéo et une carte de perspective de scène.

Tout d'abord, nous introduisons une architecture profonde entrainable de bout en bout pour le comptage de foules qui combine des représentations obtenues à l'aide de plusieurs tailles de champ récepteur et apprend l'importance de chacune de ces représentations à chaque emplacement de l'image. En d'autres termes, notre approche code de manière adaptative l'échelle des informations contextuelles requises pour prédire avec précision la densité de la foule. Cela donne un algorithme qui surpasse les méthodes de comptage de foule précédentes, en particulier lorsque les effets de perspective sont forts.

Deuxièmement, nous modélisons explicitement les changements d'échelle et la raison en termes de personnes par mètre carré. Nous montrons que fournir le modèle de perspective au réseau appris nous permet de renforcer la cohérence à l'échelle globale et que ce modèle peut être obtenu à la volée à partir des capteurs de drone. De plus, cela nous permet également d'appliquer des contraintes de cohérence temporelle qui n'ont pas à être apprises. Cela donne un algorithme qui surpasse les méthodes de pointe pour prédire la densité de foule à partir d'une caméra de drone en mouvement, en particulier lorsque les effets de perspective sont forts.

Troisièmement, pour les séquences vidéo, nous préconisons d'estimer les flux de personnes entre des images consécutives et de reconstruire les densités de personnes à partir de ces flux au lieu de les régresser directement. Cela nous permet d'imposer des contraintes beaucoup plus fortes encodant la conservation du nombre de personnes. En conséquence, il augmente considérablement les performances sans nécessiter une architecture plus complexe. De plus, cela nous permet d'exploiter la corrélation entre le flux de personnes et le flux optique pour améliorer encore les résultats. Nous montrons également que tirer parti des contraintes de conservation des personnes de manière spatiale et temporelle permet de former un modèle de comptage de foule

approfondi dans un cadre d'apprentissage actif avec beaucoup moins d'annotations. Cela réduit considérablement le coût d'annotation tout en conduisant à des performances similaires à celles du cas de supervision complète.

Mots clés : compréhension de scène, comptage de foule, réseaux de neurones profonds

Contents

List of Figures

List of Tables

1 Introduction

Understanding scenes that involve human beings is a long-lasting computer vision problem and has tremendous impact on several applications including surveillance, robotics and virtual reality. While understanding scenes and human motion is effortless for a human being, it remains challenging for a machine to estimate human-related activity.

When one wants to understand the behavior of human beings in a specific scene environment, there are several premised questions to be answered in the first place. Such as, how many people in the scene and where are they. When the people distribution is sparse and the scene structure generally contains less occlusion, the quantity and location of people can be inferred by object detection technique [76, 75, 55] with a bounding box for each person, as depicted by Fig. 1.1. However, in very crowded scenes, occlusions make detection difficult, and these approaches have been largely displaced by *counting-by-density-estimation* ones [57, 56, 59], which rely on training a regressor to estimate people density in various parts of the image and then integrating, as depicted by Fig. 1.2. For the purpose of this thesis, we will think of tackling human-centered scene understanding by density-based crowd counting which aims to model the quantity and location of people by estimating crowd density map.

Despite many years of sustained effort, crowd counting remains a difficult problem due to challenges like perspective distortion, occlusion, variability in visual appearance and shortage of data annotation. For semantic segmentation, even though many methods are proposed to reduce the requirement of data annotation, they often suffer from unstable model training due to the dependence of *adversarial-training* technique. In the face of these challenges, existing approaches are still fragile and error-prone in general unconstrained scenarios.

In this thesis, we attempt to overcome the challenge of crowd counting by learning scale-invariant features, leveraging scene geometry and enforcing temporal consistency depends on data modality. If the input data is random single image, we then learn a deep model that infer crowd density using *context-aware* features. If the input data is obtained from a drone that contains not only video sequence but also scene geometry information, we then enforce *geometric* and *physical* constraints to learn the density in physical world instead of image plane. If only video sequence is

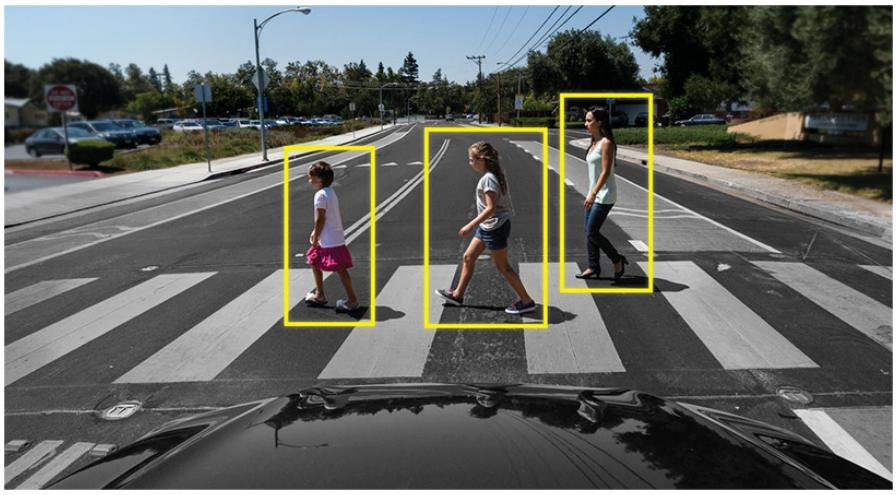
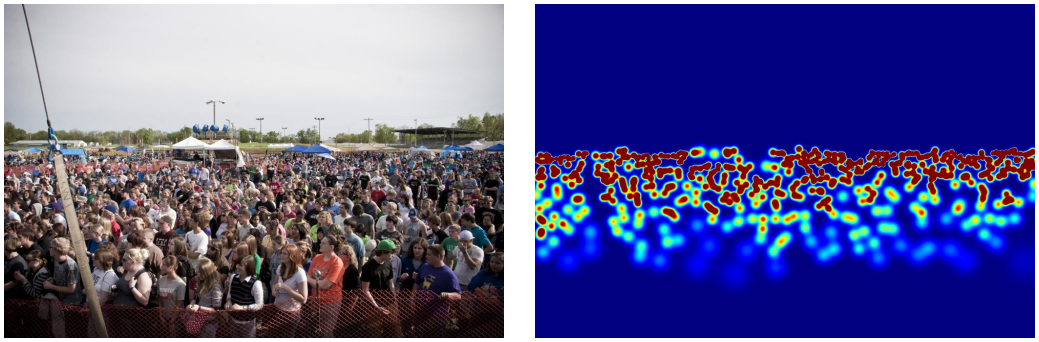


Figure 1.1 – **Pedestrian detection.** When the people are clearly isolated without much occlusion, we are able to measure the location and quantity of people using pedestrian detectors.



(a) Input image (b) Ground truth crowd density map

Figure 1.2 – **Crowd density estimation**. (a) Image with dense crowd. (b) Corresponding density map. Modeling people as crowd density is more robust to occlusion for dense crowd compared with detection-based approaches.

available without scene perspective information, we are also able to enforce temporal consistency that not only improves the crowd counting performance but also provides us a general description of crowd motion in the scene.

In the remainder of this chapter, we first define the crowd counting problem and then briefly discuss a few practical applications and present several key challenges related to this task. Finally, we summarize our main contributions and give an outline of the thesis.

1.1 Problem Definition

Our goal is to understand scenes that involve human beings, to do so, we decompose it into estimating people quantity and location by crowd counting technique.

Crowd counting is to estimate the pre-defined crowd density map from given images. To obtain the ground-truth density maps, we rely on the same strategy as previous work [46, 83, 115, 81]. Specifically, to each image, we associate a set of 2D points that denote the position of each human head in the scene. The corresponding ground-truth density map is obtained by convolving an image containing ones at these locations and zeroes elsewhere with a Gaussian kernel. Therefore the crowd counting problem can be seen as pixel-wise regression given input image and the number of people in a region is just the integrity of crowd density map within the target region. Due to perspective distortion, occlusion, variability in crowd appearance and shortage of data annotation, robust crowd counting is still a challenging task in computer vision.

1.2 Motivation and Applications

One of the most remarkable feats of the human visual system is how rapidly, accurately and comprehensively it can recognize and understand the complex visual world. Most human social activities involve visual perception of the scene, for example, simple daily walking in the street requires people to recognize the pedestrian location to avoid any collision. Often, within a single glance, humans are able to quantify and localize the people in front of them. However, it remains challenging for machines to perform similar activities. Automatic and reliable estimation of human quantity and location has therefore emerged as a pressing need for a variety of industries and finds numerous applications ranging from robotics to surveillance. In the following, we briefly discuss a few prominent applications for crowd counting.

Crowd counting is important for many applications in the industry. We hereby list a few examples as below:

Security and Surveillance. Event security is an imperative measure to ensure that public events remain under control at all times. For event like a football game, many people sit next to each other in a generally tiny space, therefore it is very likely to cause stampede accident without properly crowd density monitor. With the help of crowd counting technique, the crowd number and location can be monitored in real time without any effort. This is much more reliable than human monitor which is often suffers from fatigue and limited vision scope.

Traffic Control. Estimating crowd density can be used to optimize the schedule of traffic light. Specifically, if there are many people waiting there, the traffic light would expect to be green while it should be red when no one is there. In comparison, a fixed schedule of traffic light is extremely inefficient especially during the peak hour.

Autonomous Landing of Drones. Automatic control system for drones requires the drone to take off and land in the proper place without leveraging human effort. With the help of crowd counting technique, a drone can choose the right place to land on and therefore can avoid potential collision accident.

(a) (b)

Figure 1.3 – **Challenges in Crowd Counting.** **(a)** Perspective distortion. People far away look much smaller than the close one in image plane even though they have similar size in real world. **(b)** Occlusion. For people in the front, we can see the human body while people behind is largely occluded and we can only see the head regions.

1.3 Challenges

Even though Deep Nets technique dramatically boost the performance of many computer vision tasks, robust crowd counting is still a challenging task due to many reasons. Fig. 1.3 depicts some factors that limit the performance of crowd counting. We discuss more details of these challenges below and describe common ways to address them.

Perspective Distortion. In photography and cinematography, perspective distortion is a warping or transformation of an object and its surrounding area that differs significantly from what the object would look like with a normal focal length, due to the relative scale of nearby and distant features. Perspective distortion is determined by the relative distances at which the image is captured and viewed, and is due to the angle of view of the image being either wider or narrower than the angle of view at which the image is viewed, hence the apparent relative distances differing from what is expected. Due to perspective distortion, people far away look smaller than the close one, as depicted by Fig. 1.3 (a), which makes it difficult to localize and count the people with different sizes in image plane. The common approach to solve this problem is through image pyramid which resizes input image into different resolution so that the model can learn scale-invariant features given images from multiple resolutions.

Occlusion. Occlusion is extremely common in dense crowd. As depicted by Fig. 1.3 (b), we can see the body part for people in the front while it is not the case for people far away as they are occluded by other people. Therefore, single image crowd counting often suffer from the occlusion problem, while this can be largely eased by video sequence data where people occluded at current frame may appear in previous or future frames.

1.4 Contributions

The main goal of this thesis is to develop algorithms for efficient crowd counting given different input modalities. Precisely, the input modality includes single image, aerial video sequence with geometry information and surveillance video without geometry information. We describe below the main contributions of this thesis.

Single image Crowd Counting. We introduce a deep architecture that explicitly extracts features over multiple receptive field sizes and learns the importance of each such feature at every image location, thus accounting for potentially rapid scale changes. In other words, our approach adaptively encodes the scale of the contextual information necessary to predict crowd density.

Crowd Counting with Aerial Videos. We introduce a crowd density estimation method that *explicitly* accounts for perspective distortion to produce a real-world density map, as opposed to an image-based one. To this end, it takes advantage of the fact that drone cameras can be naturally registered to the scene using the drone's internal sensors, which as we will see are accurate enough for our purposes.

Crowd Counting with Surveillance Videos. We introduce a novel flow-based approach to estimating people densities from video sequences that enforces strong temporal consistency constraints without requiring complex network architectures. Not only does it boost performance, it also makes it possible to implement an active-learning approach that leverages the expected consistency to reduce sixteen-fold the required amount of annotated data while preserving accuracy.

1.5 Outlines

The remainder of this thesis is organized as follows. In chapter 2, we briefly summarize recent related work in crowd counting task. In chapter 3 we introduce a general single image crowd counting architecture. Chapter 4 presents our approach to count people from aerial videos where the sensors provide us not only video sequence but also scene geometry information. Chapter 5 introduces a general approach to count people in video sequence which captures temporal consistency among video frames. Finally, Chapter 6 concludes the thesis with a short summary and brief discussion of future research directions. The content from Chapter 3 to Chapter 5 is already published [57, 56, 59, 58] as part of my PhD study.

2 Related Work

Given a single image of a crowded scene, the currently dominant approach to counting people is to train a deep network to regress a people density estimate at every image location. This density is then integrated to deliver an actual count [64, 86, 57, 56, 39, 101, 48, 109, 52, 107, 108, 92, 21, 100, 7, 110, 40, 34, 62, 51, 93, 63, 117, 103]. In this section, we first review these approaches and then introduce existing attempts at reducing the amount of supervision they require.

2.1 Single Image Crowd Counting.

Early crowd counting methods [105, 104, 47] tended to rely on *counting-by-detection*, that is, explicitly detecting individual heads or bodies and then counting them. Unfortunately, in very crowded scenes, occlusions make detection difficult, and these approaches have been largely displaced by *counting-by-density-estimation* ones, which rely on training a regressor to estimate people density in various parts of the image and then integrating. This trend began in [18, 44, 26], using either Gaussian Process or Random Forests regressors. Even though approaches relying on low-level features [20, 17, 13, 73, 18, 36] can yield good results, they have now mostly been superseded by CNN-based methods [115, 83, 16], a survey of which can be found in [91]. The same can be said about methods that count objects instead of people [4, 5, 19].

The people density we want to measure is the number of people per unit area *on the ground*. However, the deep nets operate in the image plane and, as a result, the density estimate can be severely affected by the local scale of a pixel, that is, the ratio between image area and corresponding ground area. This problem has long been recognized. For example, the algorithms of [111, 42] use geometric information to adapt the network to different scene geometries. Because this information is not always readily available, other works have focused on handling the scale implicitly within the model. In [91], this was done by learning to predict pre-defined density levels. These levels, however, need to be provided by a human annotator at training time. By contrast, the algorithms of [70, 85] use image patches extracted at multiple scales as input to a multi-stream network. They then either fuse the features for final density prediction [70]

without accounting for continuous scale changes or introduce an *ad hoc* term in the training loss function [85] to enforce prediction consistency across scales. This, however, does not encode contextual information into the features produced by the network and therefore has limited impact. While [115, 16] aim to learn multi-scale features, by using different receptive fields, they combine all of these features to predict the density.

In other words, while the previous methods account for scale, they ignore the fact that the suitable scale varies smoothly over the image and should be handled adaptively. This was addressed in [41] by weighting different density maps generated from input images at various scales. However, the density map at each scale only depends on features extracted at this particular scale, and thus may already be corrupted by the lack of adaptive-scale reasoning. Here, we argue that one should rather extract *features* at multiple scales and learn how to adaptively combine them. While this, in essence, was also the motivation of [83, 81], which train an extra classifier to assign the best receptive field for each image patch, these methods remain limited in several important ways. First, they rely on classifiers, which requires pre-training the network before training the classifier, and thus is not end-to-end trainable. Second, they typically assign a single scale to an *entire* image patch that can still be large and thus do not account for rapid scale changes. Last, but not least, the range of receptive field sizes they rely on remains limited in part because using much larger ones would require using much deeper architectures, which may not be easy to train given the kind of networks being used.

By contrast, in this thesis, we introduce an end-to-end trainable architecture that adaptively fuses multi-scale features, without explicitly requiring defining patches, but rather by learning how to weigh these features for each individual pixel, thus allowing us to accommodate rapid scale changes. By leveraging multi-scale pooling operations, our framework can cover an arbitrarily large range of receptive fields, thus enabling us to account for much larger context than with the multiple receptive fields used by the above-mentioned methods. In Section 3.2, we will demonstrate that it delivers superior performance.

2.2 Handling Perspective Distortion

Earlier approaches to handling such distortions [111] involve regressing to both a crowd count and a density map. Unlike ours that passes a perspective map as an input to the deep network, they use the perspective map to compute a metric and use it to retrieve candidate training scenes with similar distortions before tuning the model. This complicates training, which is not end-to-end, and decreases performance.

These approaches were recently extended by [83], whose *SwitchCNN* exploits a classifier that greedily chooses the sub-network that yields the best crowd counting performance. Max pooling is used extensively to down-scale the density map output, which improves the overall accuracy of the counts but decreases that of the density maps as pooling incurs a loss in localization precision.

Perspective distortion is also addressed in [70] via a scale-aware model called *HydraCNN*, which uses different-sized patches as input to the CNN to achieve scale-invariance. To the same end, different kernel sizes are used in [115] and in [37] features from different layers are extracted instead. In the recent method of [91], a network dubbed CP-CNN combines local and global information obtained by learning density at different resolutions. It also accounts for density map quality by adding extra information about the pre-defined density level of different patches and images. While useful, this information is highly scene specific and would make generalization difficult. More recent works use different techniques, such as a growing CNN [81], fusing crowd counting with people detection [50], adding a new measurement between prediction and ground truth density map [16], using a scale-consistency regularizer [85], employing a pool of decorrelated regressors [88], refining the density map in an iterative process [74], leveraging web-based unlabeled data [61], to further boost performance. However, none of them is specifically designed to handle perspective effects.

In any event, all the approaches mentioned above rely on the network learning about perspective effects without explicitly modeling them. As evidenced by our results, this is suboptimal given the finite amounts of data available in practical situations. Furthermore, while learning about perspective effects to account for the varying people sizes, these methods still predict density in the image plane, thus leading to the unnatural phenomenon that real-world regions with the same number of people are assigned different densities. By contrast, we produce densities expressed in terms of number of people per square meter of ground and thus are immune to this problem.

2.3 Enforcing Temporal Consistency.

While most methods work on individual images, a few have been extended to exploit temporal consistency [78, 80, 79]. Perhaps the most popular way to do so is to use an LSTM [33]. For example, in [106], the ConvLSTM architecture [87] is used for crowd counting purposes. It is trained to enforce consistency both in the forward and the backward direction. In [114], an LSTM is used in conjunction with an FCN [65] to count vehicles in video sequences. A Locality-constrained Spatial Transformer (LST) is introduced in [25]. It takes the current density map as input and outputs density maps in the next frames. The influence of these estimates on crowd density depends on the similarity between pixel values in pairs of neighboring frames.

While effective these approaches have two main limitations. First, at training time, they can only be used to impose consistency across annotated frames and cannot take advantage of unannotated ones to provide self-supervision. Second, they do not explicitly enforce the fact that people numbers must be conserved over time, except at the edges of the field of view. The method in the previous thesis addresses both these issues. However, as will be discussed in more detail in Section 5.1.1, because the people conservation constraints are expressed in terms of numbers of people in neighboring image areas, they are much weaker than they should be.

2.4 Introducing Flow Variables.

Imposing strong conservation constraints when tracking people has been a concern long before the advent of deep learning [71, 98, 15, 49, 23, 43, 28, 14, 69, 68, 2, 11]. For example, in [11], people tracking is formulated as multi-target tracking on a grid and gives rise to a linear program that can be solved efficiently using the K-Shortest Path algorithm [96]. The key to this formulation is the use as optimization variables of people flows from one grid location to another, instead of the actual number of people in each grid location. In [72], a people conservation constraint is enforced and the global solution is found by a greedy algorithm that sequentially instantiates tracks using shortest path computations on a flow network [112]. Such people conservation constraints have since been combined with additional ones to further boost performance. They include appearance constraints [9, 24, 10] to prevent identity switches, spatio-temporal constraints to force the trajectories of different objects to be disjoint [32], and higher-order constraints [15, 23].

However, none of these methods rely on deep learning. These kind of flow constraints have therefore never been used in a deep crowd counting context and are designed for scenarios in which people can still be tracked individually. The recent approach of [77] is a good example of this. It leverages density maps and network flow constraints to improve multiple object tracking but still relies on connecting individual people detections. In this thesis, we demonstrate that this approach can also be brought to bear in a deep pipeline to handle dense crowds in which people cannot be tracked as individuals anymore.

2.5 Moving Away from Full Supervision

There are relatively few people-counting approaches that rely on self- or weak-supervision. We discuss them below and argue that they lack some of the key features of ours.

Semi-Supervised Crowd Counting. In [82], an autoencoder is used to learn most of the model parameters without supervision. Only those of the last two layers are learned with full supervision, which helps when there is very little annotated data but not when there is some more. In [67], only 10% of the annotated training images are used to pre-train a model and the algorithm relies on transfer-learning to align the feature distributions across unlabeled images with similar people counts in the remaining 90%. Unfortunately, this method depends crucially on the quality of the pre-training. If it is not good enough, the auto-annotation of the unlabeled images is likely to cause a performance drop. Furthermore, this approach still requires image pairs from different domains that feature the same number of people, which is hard to obtain in many real world cases. Finally, it only outputs the final crowd count without a density map that denotes people's locations. Several very recent work [93, 63] extend this auto-annotation technique by directly auto-annotating the crowd density map [93] or an auxiliary segmentation mask [59] based on a pre-trained model with a small amount of labeled data. As no physical world constraint is enforced in these models, the pseudo-ground truth can be very different from the true one if the labeled and unlabeled images follow different distributions.

Weakly Supervised Crowd Counting. Another way to reduce the annotation cost is to use weak supervision, as in [12]. Instead of object-wise annotation, it relies on region-wise annotation. The image is split into arbitrarily-shaped regions that each contain two or three people. A Gaussian Process is used to map images pixels to a density map. As no localization supervision is provided, the network is prone to producing uninterpretable density maps because edges, image acquisition artifacts, and tiny fluctuations in appearance can yield larger feature changes than expected. Furthermore, manually splitting the image into regions that all contain the required number of people is non-trivial and time consuming.

Self-Supervised Crowd Counting. The approach of [61, 60] is probably the one most related to ours. Two extra unlabeled datasets are collected from Google by keyword searches and query-by-example image retrieval. Then, a multi-task network is trained to rank image patches according to their crowd density, and based on the observation that any sub-image of a crowded scene image is guaranteed to contain the same number or fewer persons than the super-image. Such inequality constraints can be viewed as a weaker version of our people conservation constraints, which are equalities. However, the resulting accuracy depends on finding and properly curating the unlabeled dataset. This is a labor-intensive process because one must ensure that the unlabeled images from the internet exhibit a similar crowd density and viewpoint angle.

3 Single Image Crowd Counting

Most previous work focus on single image crowd counting, which aims to estimate the corresponding density map given random single image and standard convolutions are at the heart of these approaches. By using the same filters and pooling operations over the whole image, these implicitly rely on the same receptive field everywhere. However, due to perspective distortion, one should instead change the receptive field size across the image. In the past, this has been addressed by combining either density maps extracted from image patches at different resolutions [70] or feature maps obtained with convolutional filters of different sizes [115, 16]. However, by indiscriminately fusing information at all scales, these methods ignore the fact that scale varies continuously across the image. While this was addressed in [83, 81] by training classifiers to predict the size of the receptive field to use locally, the resulting methods are not end-to-end trainable; cannot account for rapid scale changes because they assign a single scale to relatively large patches; and can only exploit a small range of receptive fields for the networks to remain of a manageable size.

In this chapter, we introduce a deep architecture that explicitly extracts features over multiple receptive field sizes and learns the importance of each such feature at every image location, thus accounting for potentially rapid scale changes. In other words, our approach adaptively encodes the scale of the contextual information necessary to predict crowd density. This is in contrast to crowd-counting approaches that also use contextual information to account for scaling effects as in [85], but only in the loss function as opposed to computing true multi-scale features as we do. We will show that it works better on uncalibrated images. When calibration data is available, we will also show that it can be leveraged to infer suitable local scales even better and further increase performance.

Our contribution is therefore an approach that incorporates multi-scale contextual information directly into an end-to-end trainable crowd counting pipeline, and learns to exploit the right context at each image location. As shown by our experiments, we consistently outperform previous work on all standard crowd counting benchmarks, such as ShanghaiTech, WorldExpo'10, UCF_CC_50 and UCF_QNRF, as well as on our own Venice dataset, which features strong

perspective distortion.

3.1 Approach

As discussed above, we aim to exploit context, that is, the large-scale consistencies that often appear in images. However, properly assessing what the scope and extent of this context should be in images that have undergone perspective distortion is a challenge. To meet it, we introduce a new deep net architecture that adaptively encodes multi-level contextual information into the features it produces. We then show how to use these scale-aware features to regress to a final density map, both when the cameras are not calibrated and when they are.

3.1.1 Scale-Aware Contextual Features

We formulate crowd counting as regressing a people density map from an image. Given a set of N training images $\{I_i\}_{1\leq i\leq N}$ with corresponding ground-truth density maps $\{D_i^{gt}\}$, our goal is to learn a non-linear mapping $\mathcal{F}$ parameterized by θ that maps an input image I_i to an estimated density map $D_i^{est}(I_i) = \mathcal{F}(I_i, \theta)$ that is as similar as possible to D_i^{gt} in L^2 norm terms.

Following common practice [66, 76, 55], our starting point is a network comprising the first ten layers of a pre-trained *VGG-16* network [89]. Given an image I, it outputs features of the form

$$\mathbf{f}_v = \mathcal{F}_{vgg}(I) \ , \tag{3.1}$$

which we take as base features to build our scale-aware ones.

As discussed in Section 2.1, the limitation of $\mathcal{F}_{vgg}$ is that it encodes the same receptive field over the entire image. To remedy this, we compute scale-aware features by performing *Spatial Pyramid Pooling* [30] to extract multi-scale context information from the VGG features of Eq. 3.1. Specifically, as illustrated at the bottom of Fig. 3.1, we compute these scale-aware features as

$$\mathbf{s}_j = U_{bi}(\mathcal{F}_j(P_{ave}(\mathbf{f}_v, j), \theta_j)) \ , \tag{3.2}$$

where, for each scale j, $P_{ave}(\cdot, j)$ averages the VGG features into $k(j) \times k(j)$ blocks; $\mathcal{F}_j$ is a convolutional network with kernel size 1 to combine the context features across channels without changing their dimensions. We do this because SPP keeps each feature channel independent, thus limiting the representation power. We verified that without this the performance drops. This is in contrast to earlier arthitectures that convolve to reduce the dimension [97, 116]; and U_{bi} represents bilinear interpolation to up-sample the array of contextual features to be of the same size as $\mathbf{f}_v$. In practice, we use $S = 4$ different scales, with corresponding block sizes $k(j) \in \{1, 2, 3, 6\}$ since it shows better performance compared with other settings.

The simplest way to use our scale-aware features would be to concatenate all of them to the

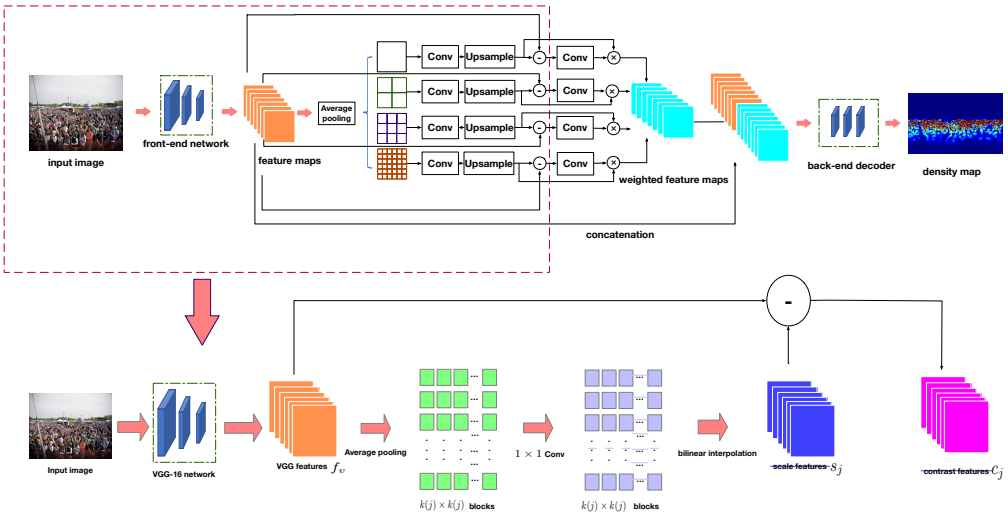


Figure 3.1 – **Context-Aware Network. (Top)** RGB images are fed to a font-end network that comprises the first 10 layers of the *VGG-16* network. The resulting local features are grouped in blocks of different sizes by average pooling followed by a 1×1 convolutional layer. They are then up-sampled back to the original feature size to form the contrast features. Contrast features are further used to learn the weights for the scale-aware features that are then fed to a back-end network to produce the final density map. **(Bottom)** As shown in this expanded version of the first part of the network, the contrast features are the difference between local features and context features.

original VGG features $\mathbf{f}_v$. This, however, would not account for the fact that scale varies across the image. To model this, we propose to learn to predict weight maps that set the relative influence of each scale-aware feature at each spatial location. To this end, we first define contrast features as

$$\mathbf{c}_j = \mathbf{s}_j - \mathbf{f}_v \ . \tag{3.3}$$

They capture the differences between the features at a specific location and those in the neighborhood, which often is an important visual cue that denotes saliency. Note that, for human beings, saliency matters. In our context, these contrast features provide us with important information to understand the local scale of each image region. We therefore exploit them as input to auxiliary networks with weights θ_{sa}^j that compute the weights ω_j assigned to each one of the S different scales we use. Each such network outputs a scale-specific weight map of the form

$$\omega_j = \mathcal{F}_{sa}^j(\mathbf{c}_j, \theta_{sa}^j) \ . \tag{3.4}$$

$\mathcal{F}_{sa}^j$ is a 1×1 convolutional layer followed by a sigmoid function to avoid division by zero. We then employ these weights to compute our final contextual features as

$$\mathbf{f}_I = \left[\mathbf{f}_v | \frac{\sum_{j=1}^S \omega_j \odot \mathbf{s}_j}{\sum_{j=1}^S \omega_j} \right] \ , \tag{3.5}$$

where $[\cdot|\cdot]$ denotes the channel-wise concatenation operation, and $\odot$ is the element-wise product

between a weight map and a feature map.

Altogether, as illustrated in Fig. 3.1, the network $\mathcal{F}(I, \theta)$ extracts the contextual features $\mathbf{f}_I$ as discussed above, which are then passed to a decoder consisting of several dilated convolutions that produces the density map. The specific architecture of the network is described in Table 3.1. As shown by our experiments, this network already outperforms previous work on all benchmark datasets, without explicitly using information about camera geometry. As discussed below, however, these results can be further improved when such information is available.

3.1.2 Geometry-Guided Context Learning

Because of perspective distortion, the contextual scope suitable for each region varies across the image plane. Hence, scene geometry is highly related to contextual information and could be used to guide the network to better adjust to the scene context it needs.

We therefore extend the previous approach to exploiting geometry information when it is available. To this end, we represent the scene geometry of image I_i with a *perspective map* M_i, which encodes the number of pixels per meter in the image plane. Note that this perspective map has the same spatial resolution as the input image. We therefore use it as input to a truncated *VGG-16* network. In other words, the base features of Eq. 3.1 are then replaced by features of the form

$$\mathbf{f}_g = \mathcal{F}'_{vgg}(M_i, \theta_g) \ , \tag{3.6}$$

where $\mathcal{F}'_{vgg}$ is a modified *VGG-16* network with a single input channel. To initialize the weights corresponding to this channel, we average those of the original three RGB channels. Note that we also normalize the perspective map M_i to lie within the same range as the RGB images. Even though this initialization does not bring any obvious difference in the final counting accuracy, it makes the network converge much faster.

To further propagate the geometry information to later stages of our network, we exploit the modified VGG features described above, which inherently contain geometry information, as an additional input to the auxiliary network of Eq. 3.4. Specifically, the weight map for each scale is then computed as

$$\omega_j = \mathcal{F}^j_{gc}([\mathbf{c}_j|\mathbf{f}_g], \theta^j_{gc}) \ . \tag{3.7}$$

These weight maps are then used as in Eq. 3.5. Fig. 3.2 depicts the corresponding architecture.

3.1.3 Training Details and Loss Function

Whether with or without geometry information, our networks are trained using the L^2 loss defined as

$$L(\theta) = \frac{1}{2B} \sum_{i=1}^{B} \|D_i^{gt} - D_i^{est}\|_2^2 \ , \tag{3.8}$$

layer	front-end($\mathcal{F}_{vgg}$)	layer	back-end decoder
1 - 2	3×3×64 conv-1	1	3×3×512 conv-2
	2 × 2 max pooling	2	3×3×512 conv-2
3 - 4	3×3×128 conv-1	3	3×3×512 conv-2
	2 × 2 max pooling	4	3×3×256 conv-2
5 - 7	3×3×256 conv-1	5	3×3×128 conv-2
	2 × 2 max pooling	6	3×3×64 conv-2
8 - 10	3×3×512 conv-1	7	1×1×1 conv-1

Table 3.1 – **Network architecture of proposed model** Convolutional layers are represented as "(kernel size) × (kernel size) × (number of filters) conv-(dilation rate)".

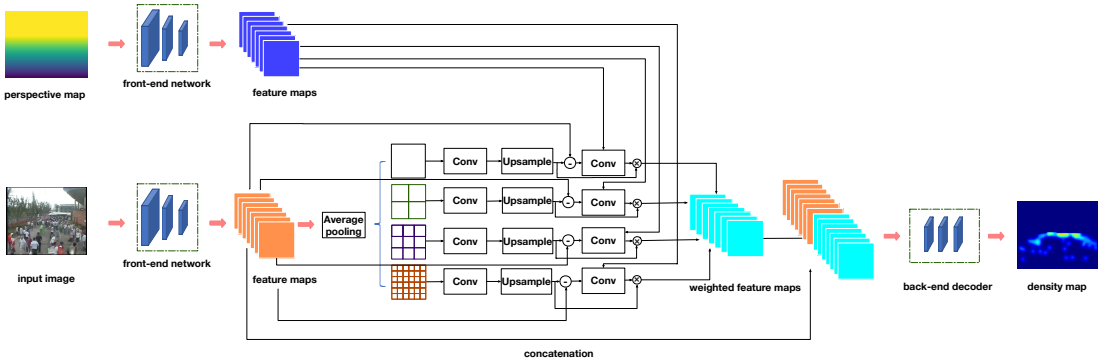


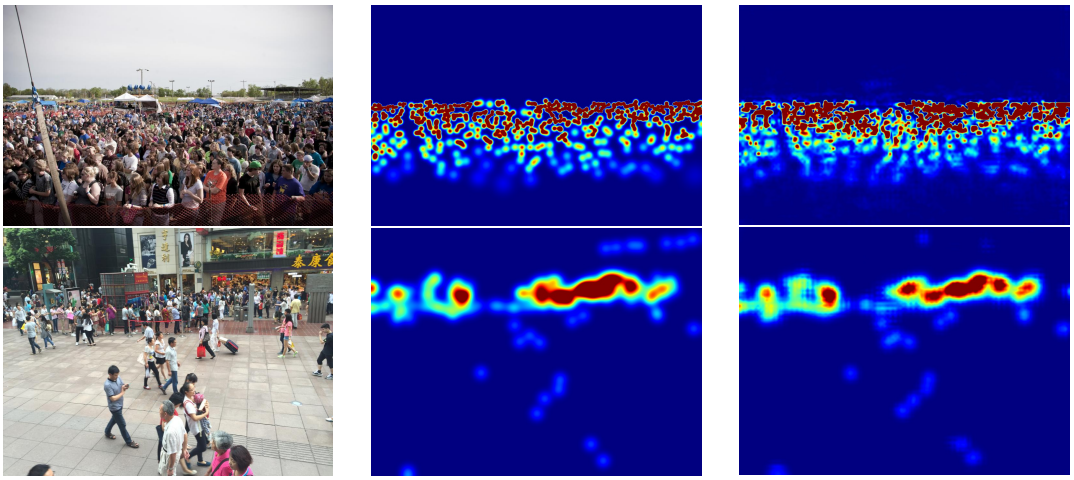
Figure 3.2 – **Expanded Context-Aware Network.** To account for camera registration information when available, we add a branch to the architecture of Fig. 3.1. It takes as input a perspective map that encodes local scale. Its output is concatenated to the original contrast features and the resulting scale-aware features are used to estimate people density.

where B is the batch size. To obtain the ground-truth density maps D_i^{gt}, we rely on the same strategy as previous work [46, 83, 115, 81]. Specifically, to each image I_i, we associate a set of c_i 2D points $P_i^{gt} = \{P_i^j\}_{1 \leq j \leq c_i}$ that denote the position of each human head in the scene. The corresponding ground-truth density map D_i^{gt} is obtained by convolving an image containing ones at these locations and zeroes elsewhere with a Gaussian kernel $\mathcal{N}^{gt}(p|\mu, \sigma^2)$ [53]. We write

$$\forall p \in I_i, D_i^{gt}(p|I_i) = \sum_{j=1}^{c_i} \mathcal{N}^{gt}(p|\mu = P_i^j, \sigma^2) \ , \tag{3.9}$$

where μ and σ represent the mean and standard deviation of the normal distribution. To produce the comparative results we will show in Section 3.2, we use the same σ as the methods we compare against.

To minimize the loss of Eq. 3.8, we use Stochastic Gradient Descent (SGD) with batch size 1 for various size dataset and Adam with batch size 32 for fixed size dataset. Furthermore, during training, we randomly crop image patches of $\frac{1}{4}$ the size of the original image at different locations.



(a) Input image (b) Ground truth (c) Our prediction

Figure 3.3 – **Crowd density estimation on ShanghaiTech**. First row: Image from Part A. Second row: Image from Part B. Our model adjusts to rapid scale changes and delivers density maps that are close to the ground truth.

These patches are further mirrored to double the training set.

3.2 Experiments

In this section, we evaluate the proposed approach. We first introduce the evaluation metrics and benchmark datasets we use in our experiments. We then compare our approach to previous methods, and finally perform a detailed ablation study.

3.2.1 Evaluation Metrics

Previous works in crowd density estimation use the mean absolute error (MAE) and the root mean squared error ($RMSE$) as evaluation metrics [115, 111, 70, 83, 106, 91]. They are defined as

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|z_i - \hat{z}_i| \text{ and } RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(z_i - \hat{z}_i)^2}\ ,$$

where N is the number of test images, z_i denotes the true number of people inside the ROI of the ith image and $\hat{z}_i$ the estimated number of people. In the benchmark datasets discussed below, the ROI is the whole image except when explicitly stated otherwise. Note that number of people can be recovered by integrating over the pixels of the predicted density maps as $\hat{z}_i = \sum_{p \in I_i} D_i^{est}(p|I_i)$.

Model	Part_A		Part_B	
	MAE	$RMSE$	MAE	$RMSE$
Zhang *et al.* [111]	181.8	277.7	32.0	49.8
MCNN [115]	110.2	173.2	26.4	41.3
Switch-CNN [83]	90.4	135.0	21.6	33.4
CP-CNN [91]	73.6	106.4	20.1	30.1
ACSCP [85]	75.7	102.7	17.2	27.4
Liu *et al.* [61]	73.6	112.0	13.7	21.4
D-ConvNet [88]	73.5	112.3	18.7	26.0
IG-CNN [81]	72.5	118.2	13.6	21.1
ic-CNN[74]	68.5	116.2	10.7	16.0
CSRNet [46]	68.2	115.0	10.6	16.0
SANet [16]	67.0	104.5	8.4	13.6
OURS-CAN	**62.3**	**100.0**	**7.8**	**12.2**

Table 3.2 – **Comparative results on the ShanghaiTech dataset.**

Model	MAE	$RMSE$
Idrees *et al.* [36]	315	508
MCNN [115]	277	426
Encoder-Decoder [6]	270	478
CMTL [90]	252	514
Switch-CNN [83]	228	445
Resnet101 [31]	190	277
Densenet201 [35]	163	226
Idrees *et al.* [37]	132	191
OURS-CAN	**107**	**183**

Table 3.3 – **Comparative results on the UCF_QNRF dataset.**

3.2.2 Benchmark Datasets and Ground-truth Data

We use five different datasets to compare our approach to recent ones. The first four were released along with recent papers and have already been used for comparison purposes since. We created the fifth one ourselves and made it publicly available as well.

ShanghaiTech [115]. It comprises 1,198 annotated images with 330,165 people in them. It is divided in part A with 482 images and part B with 716. In part A, 300 images form the training set and, in part B, 400. The remainder are used for testing purposes. For a fair comparison with earlier work [115, 85, 46, 88], we created the ground-truth density maps in the same manner as they did. Specifically, for Part A, we used the geometry-adaptive kernels introduced in [115], and for part B, fixed kernels. In Fig. 3.3, we show one image from each part, along with the ground-truth density maps and those estimated by our algorithm.

UCF-QNRF [37]. It comprises 1,535 jpeg images with 1,251,642 people in them. The training

Model	MAE	$RMSE$
Idrees *et al.*[36]	419.5	541.6
Zhang *et al.* [111]	467.0	498.5
MCNN [115]	377.6	509.1
Switch-CNN [83]	318.1	439.2
CP-CNN [91]	295.8	320.9
ACSCP [85]	291.0	404.6
Liu *et al.* [61]	337.6	434.3
D-ConvNet [88]	288.4	404.7
IG-CNN [81]	291.4	349.4
ic-CNN[74]	260.9	365.5
CSRNet [46]	266.1	397.5
SANet [16]	258.4	334.9
OURS-CAN	**212.2**	**243.7**

Table 3.4 – **Comparative results on the UCF_CC_50 dataset.**

set is made of 1,201 of these images. Unlike in **ShanghaiTech**, there are dramatic variations both in crowd density and image resolution. The ground-truth density maps were generated by adaptive Gaussian kernels as in [37].

UCF_CC_50 [36]. It contains only 50 images with a people count varying from 94 to 4,543, which makes it challenging for a deep-learning approach. For a fair comparison again, the ground-truth density maps were generated using fixed kernels and we follow the same 5-fold cross-validation protocol as in [36]: We partition the images into 5 10-image groups. In turn, we then pick four groups for training and the remaining one for testing. This gives us 5 sets of results and we report their average.

WorldExpo'10 [111]. It comprises 1,132 annotated video sequences collected from 103 different scenes. There are 3,980 annotated frames, with 3,380 of them used for training purposes. Each scene contains a Region Of Interest (ROI) in which people are counted. The bottom row of Fig. 3.4 depicts three of these images and the associated camera calibration data. We generate the ground-truth density maps as in our baselines [83, 46, 16]. As in previous work [111, 115, 83, 81, 46, 16, 53, 91, 85, 74, 88] on this dataset, we report the *MAE* of each scene, as well as the average over all scenes.

Venice. The four datasets discussed above have the advantage of being publicly available but do not contain precise calibration information. In practice, however, it can be readily obtained using either standard photogrammetry techniques or onboard sensors, for example when using a drone to acquire the images. To test this kind of scenario, we used a cellphone to film additional sequences of the Piazza San Marco in Venice, as seen from various viewpoints on the second floor of the basilica, as shown in the top two rows of Fig. 3.4. We then used the white lines on the ground to compute camera models. As shown in the bottom two rows of Fig. 3.4, this yields a more accurate calibration than in **WorldExpo'10**. The resulting dataset contains 4 different

Model	Scene1	Scene2	Scene3	Scene4	Scene5	Average
Zhang *et al.* [111]	9.8	14.1	14.3	22.2	3.7	12.9
MCNN [115]	3.4	20.6	12.9	13.0	8.1	11.6
Switch-CNN [83]	4.4	15.7	10.0	11.0	5.9	9.4
CP-CNN [91]	2.9	14.7	10.5	10.4	5.8	8.9
ACSCP [85]	2.8	14.05	9.6	8.1	2.9	7.5
IG-CNN [81]	2.6	16.1	10.15	20.2	7.6	11.3
ic-CNN[74]	17.0	12.3	9.2	8.1	4.7	10.3
D-ConvNet [88]	**1.9**	12.1	20.7	8.3	**2.6**	9.1
CSRNet [46]	2.9	11.5	**8.6**	16.6	3.4	8.6
SANet [16]	2.6	13.2	9.0	13.3	3.0	8.2
DecideNet [50]	2.0	13.14	8.9	17.4	4.75	9.23
OURS-CAN	2.9	12.0	10.0	**7.9**	4.3	**7.4**
OURS-ECAN	2.4	**9.4**	8.8	11.2	4.0	**7.2**

Table 3.5 – **Comparative results in MAE terms on the WorldExpo'10 dataset.**

Model	MAE	$RMSE$
MCNN [115]	145.4	147.3
Switch-CNN [83]	52.8	59.5
CSRNet[46]	35.8	50.0
OURS-CAN	23.5	38.9
OURS-ECAN	**20.5**	**29.9**

Table 3.6 – **Comparative results on the Venice dataset.**

sequences and in total 167 annotated frames with fixed 1,280 $\times$ 720 resolution. 80 images from a single long sequence are taken as training data, and we use the images from the remaining 3 sequences for testing purposes. The ground-truth density maps were generated using fixed Gaussian kernels as in part B of the **ShanghaiTech** dataset.

3.2.3 Comparing against Recent Techniques

In Tables 3.2, 3.3, 3.4, and 3.5, we compare our results to those of the method that returns the best results for each one of the 4 public datasets, as currently reported in the literature. They are

Model	MAE	$RMSE$
VGG-SIMPLE	68.0	113.4
VGG-CONCAT	63.4	108.7
VGG-NCONT	63.1	106.4
OURS-CAN	**62.3**	**100.0**

Table 3.7 – **Ablation study on the ShanghaiTech part A dataset.**

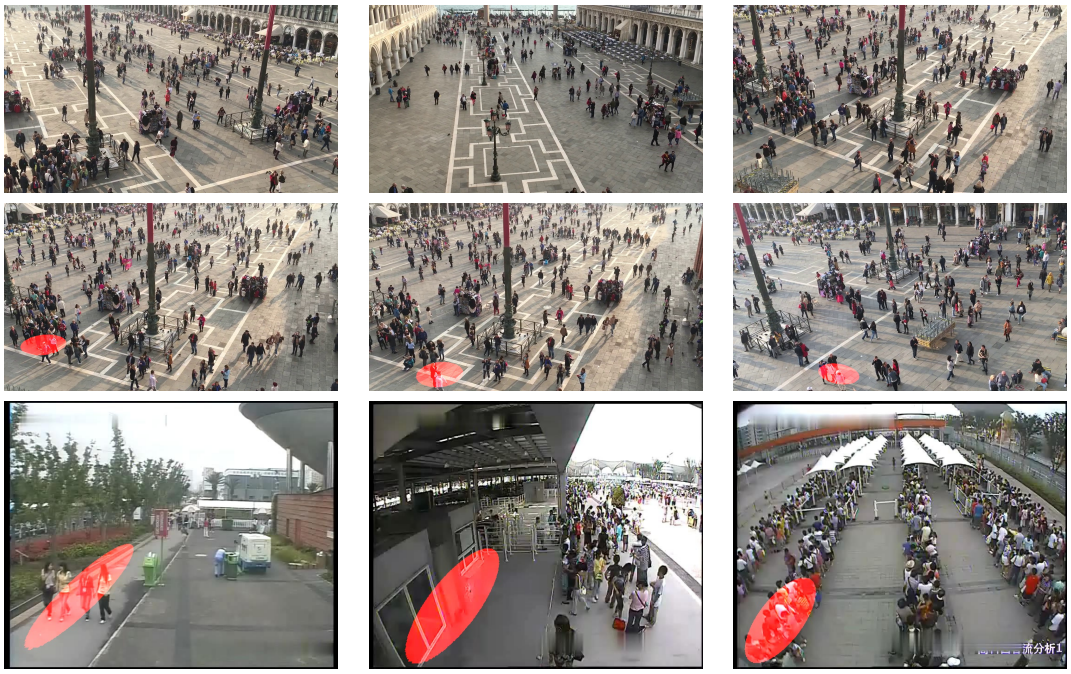


Figure 3.4 – **Calibration in Venice and WorldExpo'10.** (Top row) Images of Piazza San Marco taken from different viewpoints. (Middle row) We used the regular ground patterns to accurately register the cameras in each frame. The red ellipse overlaid in red is the projection of a 1m radius circle from the ground plane to the image plane. (Bottom row) The same 1m radius circle overlaid on three WorldExpo'10 images. As can be seen in the bottom right image, the ellipse surface corresponds to an area that could be filled by many more people that could realistically fit in a 1m radius circle. By contrast, the ellipse deformations are more consistent and accurate for Venice, which denotes a better registration.

those of [16], [37], [16], and [85], respectively. In each case, we reprint the results as given in these papers and add those of **OURS-CAN**, that is, our method as described in Section 3.1.1. On the first three datasets, we consistently and clearly outperform all other methods. On the **WorldExpo'10** dataset, we also outperform them on average, but not in every scene. More specifically, in Scenes 2 and 4 that are crowded, we do very well. By contrast, the crowds are far less dense in Scenes 1 and 5. This makes context less informative and our approach still performs honorably but looses its edge compared to the others. Interestingly, as can be seen in Table 3.5, in such uncrowded scenes, a detection-based method such as DecideNet [53] becomes competitive whereas it isn't in the more crowded ones. In Fig. 3.5, we use a **Venice** image to show how well our approach does compared to the others in the crowded parts of the scene.

The first three datasets do not have any associated camera calibration data, whereas **WorldExpo'10** comes with a rough estimation of the image plane to ground plane homography and **Venice** with an accurate one. We therefore used these homographies to run **OURS-ECAN**, our method as described in Section 3.1.2. We report the results in Tables 3.5 and 3.6. Unsurprisingly, **OURS-ECAN** clearly further improves on **OURS-CAN** when the calibration data is accurate as for **Venice** and even when it is less so as for **WorldExpo**, but by a smaller margin.

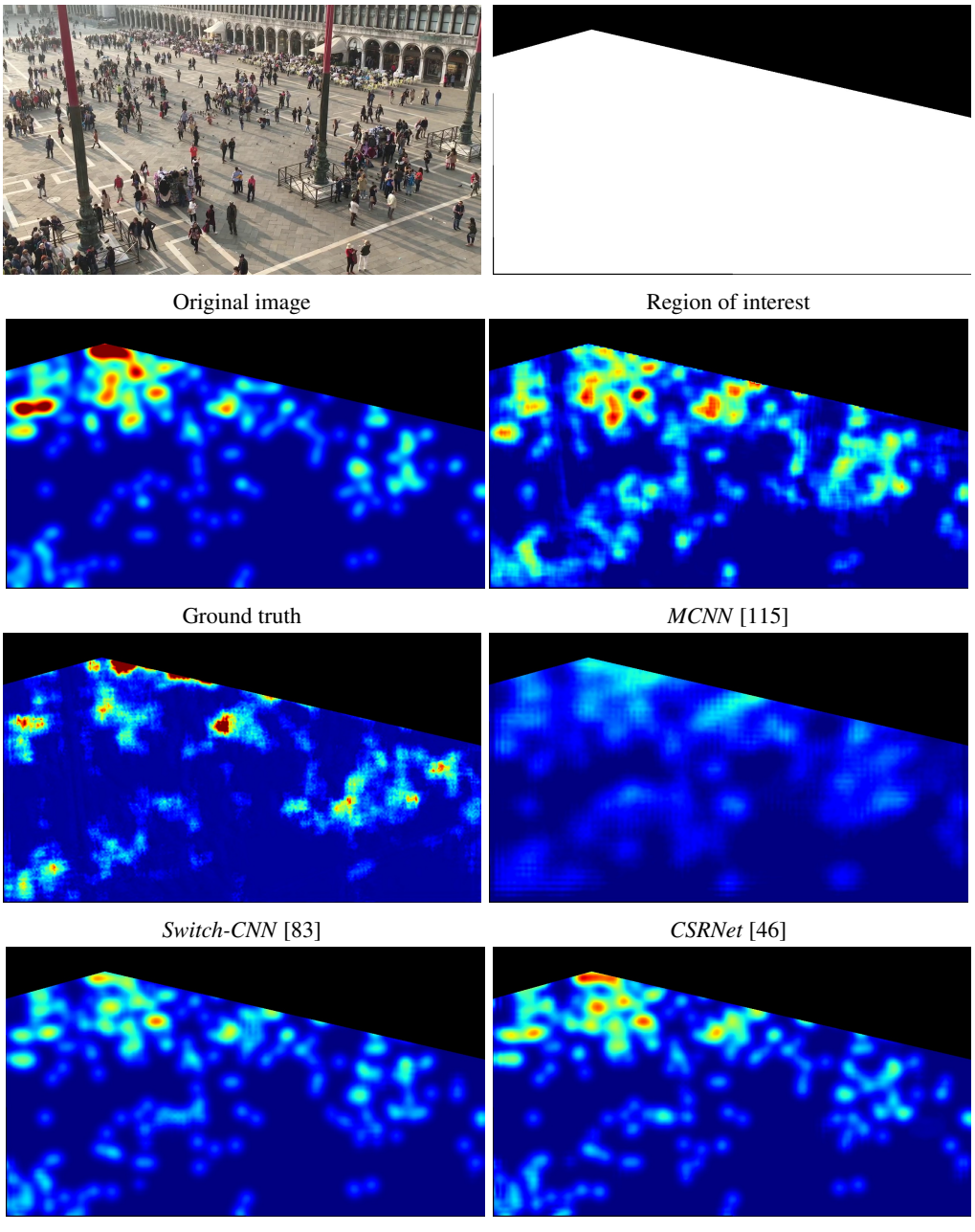


Figure 3.5 – **Density estimation in Venice.** Original image, ROI, ground truth density map within the ROI, and density maps estimated both by the baselines and our method. Note how much more similar the density map produced by **OURS-ECAN** is to the ground truth than the others, especially in the upper corner of the ROI, where people density is high.

3.2.4 Ablation Study

Finally, we perform an ablation study to confirm the benefits of encoding multiple level contextual information and of introducing contrast features.

Concatenating and Weighting VGG Features. We compare our complete model without geometry, **OURS-CAN**, against two simplified versions of it. The first one, **VGG-SIMPLE**, directly uses VGG-16 base features $\mathbf{f}_v$ as input to the decoder subnetwork. In other words, it does not adapt for scale. The second one, **VGG-CONCAT**, concatenates all scale-aware features $\{\mathbf{s}_j\}_{1\leq j\leq S}$ to the base features instead of computing their weighted linear combination, and then passes the resulting features to the decoder.

We compare these three methods on the **ShanghaiTech** Part A, which has often been used for such ablation studies [91, 16, 46]. As can be seen in Table 3.7, concatenating the VGG features as in **VGG-CONCAT** yields a significant boost, and weighing them as in **OURS-CAN** a further one.

Contrast Features. To demonstrate the importance of using contrast features to learn the network weights, we compare **OURS-CAN** against **VGG-NCONT** that uses the scale features $\mathbf{s}_j$ instead of the contrast ones to learn the weight maps. As can be seen in Table 3.7, this also results in a substantial performance loss.

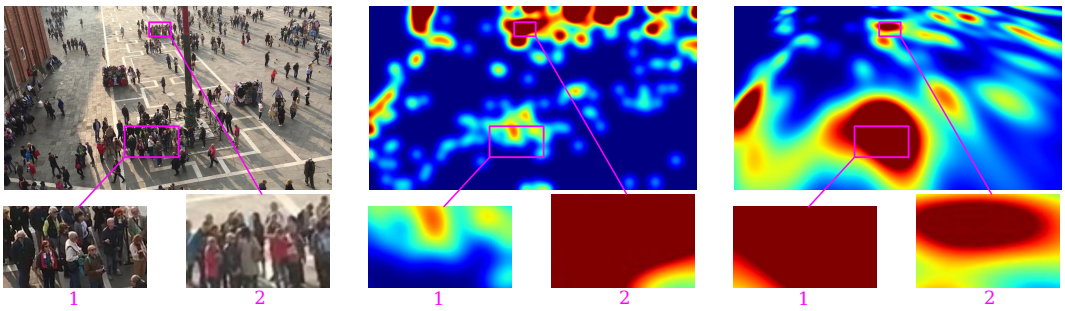
4 Crowd Counting with Aerial Videos

With the growing prevalence of drones, drone-based crowd density estimation becomes increasingly relevant to applications such as autonomous landing and video surveillance. In recent years, the emphasis has been on developing *counting-by-density* algorithms that rely on regressors trained to estimate the density of crowd per unit area so that the total numbers of people can be obtained by integration, without explicit detection being required. The regressors can be based on Random Forests [44], Gaussian Processes [18], or more recently Deep Nets [111, 115, 70, 83, 106, 91, 85, 50, 45, 81, 88, 61, 37, 74, 16], with most state-of-the-art approaches now relying on the latter.

While effective, these algorithms all estimate density in the image plane. As a consequence, and as can be seen in Fig. 4.1(a,b), two regions of the scene containing the same number of people per square meter can be assigned different densities. However, for the purposes of autonomous landing or crowd size estimation, the density of people on the ground is a more relevant measure and is *not* subject to such distortions, as shown in Fig. 4.1(c).

In this chapter, we therefore introduce a crowd density estimation method that *explicitly* accounts for perspective distortion to produce a real-world density map, as opposed to an image-based one. To this end, it takes advantage of the fact that drone cameras can be naturally registered to the scene using the drone's internal sensors, which as we will see are accurate enough for our purposes. This contrasts with methods that *implicitly* deal with perspective effects by either learning scale-invariant features [115, 83, 91] or estimating density in patches of different sizes [70]. Unlike these, we model perspective distortion globally and account for the fact that people's projected size changes consistently across the image. To this end, we feed to our density-estimation CNN not only the original image but also an identically-sized image that contains the local scale, which is a function of the camera orientation with respect to the ground plane.

An additional benefit of reasoning in the real world is that we can encode physical constraints to model the motion of people in a video sequence. Specifically, given a short sequence as input to our network, we impose temporal consistency by forcing the densities in the various images to correspond to physically possible people flows. In other words, we *explicitly* model the



(a) (b) (c)

Figure 4.1 – **Measuring people density.** (a) An image of Piazza San Marco in Venice. The two purple boxes highlight patches in which the crowd density per square meter is similar. (b) Ground-truth *image density* obtained by averaging the head annotations in the image plane. The two patches are in the same locations as in (a). The density per square pixel strongly differs due to perspective distortion: the farther patch 2 wrongly features a higher density than closer patch 1, even though the people do not stand any closer to each other. (c) By contrast the ground-truth *head plane density* introduced in Section 4.1.1 is unaffected by perspective distortion. The density in the two patches now has similar peak values, as it should.

motion of people, with physically-justified constraints, instead of *implicitly* learning long-term dependencies only across annotated frames, which are typically sparse over time, via LSTMs, as is commonly done in the literature [106].

Our contribution is therefore an approach that incorporates geometric and physical constraints directly into an end-to-end learning formalism for crowd counting using information directly obtained from the drone sensors. As evidenced by our experiments, this enables us to outperform previous work on a drone-based video sequences with severe perspective distortion.

4.1 Perspective Distortion

All existing approaches estimate the crowd density in the image plane and in terms of people per square pixel, which changes across the image even if the true crowd density per square meter is constant. For example, in many scenes such as the one of Fig. 4.1(a), the people density in farther regions is higher than that in closer regions, as can be seen in Fig. 4.1(b).

In this chapter, we train the system to directly predict the crowd density in the physical world, which does not suffer from this problem and is therefore unaffected by perspective distortion, assuming that people are standing on an approximately flat surface. Our approach could easily be extended to a non flat one given a terrain model. In a crowded scene, people's heads are more often visible than their feet. Consequently, it is a common practice to provide annotations in the form of a dot on the head for supervision purposes. To account for this, we define a *head plane*, parallel to the ground and lifted above it by the average person's height. We assume that the camera has been calibrated so that we are given the homography between the image and the head

plane.

4.1.1 Image Plane versus Head Plane Density

Let $\mathbf{H}_i$ be the homography from an image I_i to its corresponding head plane. We define the ground-truth density as a sum of Gaussian kernels centered on peoples' heads in the head plane. Because we work in the physical world, we can use the same kernel size across the entire scene and across all scenes. A head annotation P_i, that is, a 2D image point expressed in projective coordinates, is mapped to $\mathbf{H}_i P_i$ in the head plane. Given a set $A_i = \{P_i^1, ..., P_i^{c_i}\}$ of c_i such annotations, we take the *head plane density* G_i' at point P' expressed in head plane coordinates to be

$$G_i'(P') = \sum_{j=1}^{c_i} \mathcal{N}(P'; \mathbf{H}_i P_i^j, \sigma) \ , \tag{4.1}$$

where $\mathcal{N}(.; \mu, \sigma)$ is a 2D Gaussian kernel with mean μ and variance σ. We can then map this head plane density to the image coordinates, which yields a density at pixel location P given by

$$G_i(P) = G_i'(\mathbf{H}_i P) \ . \tag{4.2}$$

An example density G_i is shown in Fig. 4.1(c). Note that, while the density is Gaussian in the head plane, it is *not* in the image plane.

4.1.2 Geometry-Aware Crowd Counting

Since the head plane density map can be transformed into an image of the same size as that of the original image, we could simply train a deep network to take a 3-channel RGB image as input and output the corresponding density map. However, this would mean neglecting the geometry encoded by the ground plane homography, namely the fact that the local scale does not vary arbitrarily across the image and must remain globally consistent.

To account for this, we associate to each image I a *perspective map* M of the same size as I containing the local scale of each pixel, that is, the factor by which a small area around the pixel is multiplied when projected to the head plane. We then use a convolutional network with 4 input channels instead of only 3. The first three are the usual RGB channels, while the fourth contains the perspective map. We will show in the result section that this substantially increases accuracy over using the RGB channels only. This network is one of the *spatial streams* depicted by Fig. 4.2. To learn its weights Θ, we minimize the *head plane loss* $L_H(I, M, G; \Theta)$, which we take to be the mean square error between the predicted head plane density and the ground-truth one.

To compute the perspective map M, let us first consider the image pixel $(x, y)^\intercal$ and an infinitesimal area $dx\ dy$ surrounding it. Let $(x', y')^\intercal$ and $dx'dy'$ be their respective projections on the head plane. We take $M(x, y)$, the scale at $(x, y)^\intercal$, to be $(dx'dy')/(dx\ dy)$, which we compute as

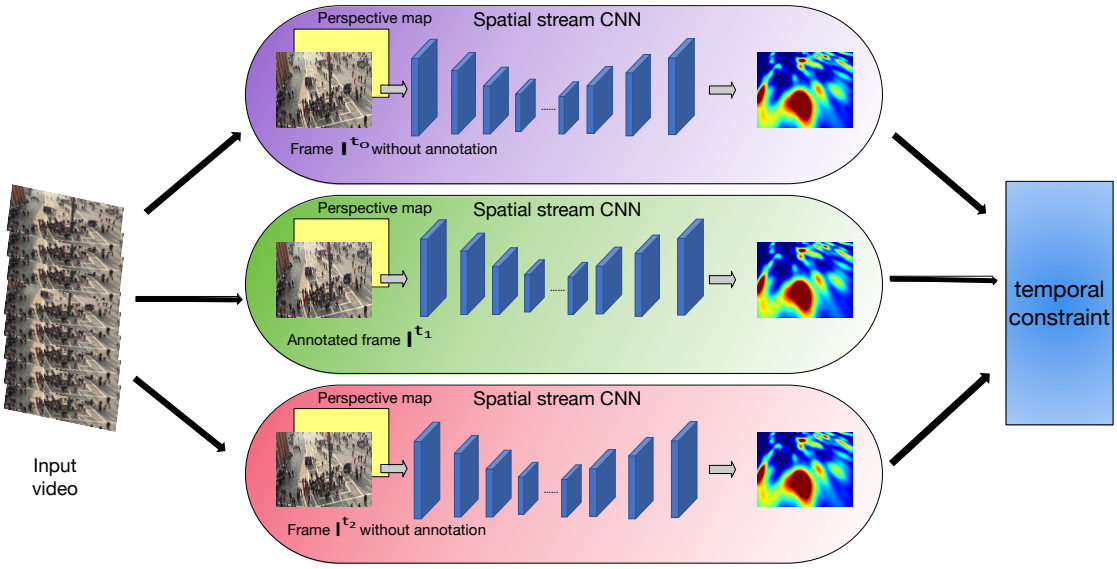


Figure 4.2 – **Three-stream architecture.** A spatial stream is a CSRNet [45] with 3 transposed convolutional layers, that takes as input the image and a perspective map. It is duplicated three times to process images taken at different times and minimize a loss that enforces temporal consistency constraints.

follows. Using the variable substitution equation, we write

$$dx'dy' = |\det(J(x,y))|dx\ dy\ , \tag{4.3}$$

where $J(x,y)$ is the Jacobian matrix of the coordinate transformation at the point $(x,y)^\intercal$:

$$J = \begin{bmatrix} \frac{\partial x'}{\partial x} & \frac{\partial x'}{\partial y} \\ \frac{\partial y'}{\partial x} & \frac{\partial y'}{\partial y} \end{bmatrix} . \tag{4.4}$$

The scale map M is therefore equal to

$$M(x,y) = |\det(J(x,y))|. \tag{4.5}$$

The detailed solution can be found in [22]. Eq. 4.5 enables us to compute the perspective map that we use as an input to our network, as discussed above. It also allows us to convert between people density F in image space, that is, people per square pixel, and people density G' on the head plane. More precisely, let us consider a surface element dS in the image around point $(x,y)^\intercal$. It is scaled by $\mathbf{H}$ into $dS' = M(x,y)dS$. Since the projection does not change the number of people, we have

$$\begin{aligned} F(x,y)dS &= G'(x',y')dS' = G'(x',y')M(x,y)dS \\ \Rightarrow F(x,y) &= M(x,y)G'(x',y')\ . \end{aligned} \tag{4.6}$$

Expressed in image coordinates, this becomes

$$F(x, y) = M(x, y)G(x, y) \ , \tag{4.7}$$

which we use in the results section to compare our algorithm that produces head plane densities against the baselines that estimate image plane densities.

4.1.3 Obtaining scene geometry from UAV sensors

We calculate the homography matrix $\mathbf{H}$ using the camera's altitude h and pitch angle θ reported by the UAV sensors. We choose the world coordinate frame such that the head plane is given by $Z = 0$ and the origin $(0, 0, 0)^\intercal$ is directly under the UAV. The camera extrinsics are described by the rotation matrix $R = R_y(\frac{\pi}{2} + \theta)$ and translation vector $t = (0, 0, h)^\intercal$.

The relation between a point $(x_h, y_h, 0)^\intercal$ on the head plane and its projection $(u, v)^\intercal$ onto the image is expressed by the following equation, in homogenous coordinates:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_1 \\ R_{21} & R_{22} & R_{23} & t_2 \\ R_{31} & R_{32} & R_{33} & t_3 \end{bmatrix} \begin{bmatrix} x_h \\ y_h \\ 0 \\ 1 \end{bmatrix} , \tag{4.8}$$

where K is the camera's intrinsic matrix and $w \neq 0$ is an arbitrary scale factor. Solving for $(x_h, y_h)^\intercal$ we obtain:

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = w \left(K \begin{bmatrix} R_{11} & R_{12} & t_1 \\ R_{21} & R_{22} & t_2 \\ R_{31} & R_{32} & t_3 \end{bmatrix} \right)^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} . \tag{4.9}$$

The transformation from the image to the head plane is therefore given by the homography $\mathbf{H} = \left(K \left[\ R_1 \ \middle| \ R_2 \ \middle| \ t \ \right] \right)^{-1}$.

4.2 Temporal Consistency

The spatial stream network depicted at the top of Fig. 4.2 operates on single frames of a video sequence. To increase robustness, we now show how to enforce temporal consistency across triplets of frames. Unlike in an LSTM-based approach, such as [106], we can do this across any three frames instead of only across annotated frames. Furthermore, by working in the real world plane instead of the image plane, we can explicitly exploit physical constraints on people's motion.

4.2.1 People Conservation

An important constraint is that people do not appear or disappear from the head plane except at the edges or at specific points that can be marked as exits or entrances. To model this, we partition the head plane into K blocks. Let $N(k)$ for $1 \leq k \leq K$ denote the neighborhood of block B_k, including B_k itself. Let m_k^t be the number of people in B_k at time t and let $t_0 < t_1 < t_2$ be three different time instants. In experiments, we empirically set the block size to 30 by 30 pixels.

If we take the blocks to be large enough for people not be able to traverse more than one block between two time instants, people in the interior blocks can only come from a block in $N(k)$ at the previous instant and move to a block in $N(k)$ at the next. As a consequence, we can write

$$\forall k \quad m_k^{t_1} \leq \sum_{i \in N(k)} m_i^{t_0} \text{ and } m_k^{t_1} \leq \sum_{i \in N(k)} m_i^{t_2} \ . \tag{4.10}$$

In fact, an even stronger equality constraint could be imposed as in [11] by explicitly modeling people flows from one block to the next with additional variables predicted by the network. However, not only would this increase the number of variables to be estimated, but it would also require enforcing hard constraints between different network's outputs.

In practice, since our networks output head plane densities, we write

$$m_k^t = \sum_{(x',y')^\intercal \in B_k} \hat{G'}^t(x', y') \ , \tag{4.11}$$

where $\hat{G'}^t$ is the predicted people density at time t, as defined in Section 4.1.2. This allows us to reformulate the constraints of Eq. 4.10 in terms of densities.

4.2.2 Siamese Architecture

To enforce these constraints, we introduce the siamese architecture depicted by Fig. 4.2, with weights Θ. It comprises three identical streams, each stream is a CSRNet [45] with 3 transposed convolutional layers added before the last convolutional layer, so that the input image and output density map have the same size. These three identical steams take as input images acquired at times t_0, t_1, and t_2 along with their corresponding perspective maps, as described in Section 4.1.2. Each one produces a head plane density estimate G'^{t_i} and we define the temporal loss term $L_T(I^{t_0}, I^{t_1}, I^{t_2}, M^{t_0}, M^{t_1}, M^{t_2}; \Theta)$ as

$$\frac{1}{2K} \sum_{k=1}^{K} [(max(0, m_k^{t_1} - U_k^{t_0}))^2 + (max(0, m_k^{t_1} - U_k^{t_2}))^2] \ , \tag{4.12}$$

where m_k^t is the sum of predicted densities in block B_k, as in Eq. 4.11, and $U_k^t = \sum_{i \in N(k)} m_i^t$ is the sum of densities in the neighborhood of B_k.

In other words, L_T penalizes violations of the constraints of Eq. 4.10. At training time, we

Model	MAE	$RMSE$	$MPAE$
CSRNet [45]	50.1	54.2	125.6
MCNN [115]	23.5	30.6	143.9
SwitchCNN [83]	91.0	120.5	330.1
OURS-NoGeom	29.2	34.8	131.2
OURS-GeomOnly	20.1	24.7	135.1
OURS-Geom-Phy (frame interval 1)	**11.9**	**15.1**	116.9
OURS-Geom-Phy (frame interval 5)	16.1	20.2	**113.2**
OURS-Geom-Phy (frame interval 10)	13.4	17.2	126.2

Table 4.1 – Comparative results in terms of head plane crowd density on the **Campus** dataset.

Model	MAE	$RMSE$	$MPAE$
CSRNet [45]	51.3	57.6	126.4
MCNN [115]	24.2	37.1	146.2
SwitchCNN [83]	91.7	122.1	340.7
OURS-NoGeom	29.8	35.2	132.0
OURS-GeomOnly	21.2	24.7	136.8
OURS-Geom-Phy (frame interval 1)	**12.3**	**16.0**	117.3
OURS-Geom-Phy (frame interval 5)	16.9	22.3	**114.1**
OURS-Geom-Phy (frame interval 10)	14.2	18.0	128.7

Table 4.2 – Comparative results in terms of image plane crowd density on the **Campus** dataset.

minimize the composite loss

$$L_H(I^{t_1}, M^{t_1}, G'^{t_1}; \Theta) + L_T(I^{t_0}, I^{t_1}, I^{t_2}, M^{t_0}, M^{t_1}, M^{t_2}; \Theta), \tag{4.13}$$

where L_H is the head plane loss introduced in Section 4.1.2. Since the loss requires the ground truth density only for frame I^{t_1}, we only need annotations for that frame. Therefore, we can use arbitrarily-spaced and unannotated frames to impose temporal consistency and improve robustness, which is not something LSTM-based methods can do.

4.3 Experiments

4.3.1 Datasets and Experimental Setup

Our approach is designed to handle perspective effects as well as to enforce temporal consistency. As there is no publicly available drone-based crowd counting dataset, we filmed a six-minute long sequence using a DJI phantom 4 pro drone flying over a university campus and filming it from many different perspectives. We manually annotated 90 images such as the one of Fig. 4.3 and used 54 of them for training and validation purposes and the remainder for testing. The

Model	MAE	$RMSE$	$MPAE$
CSRNet [45]	38.5	42.7	121.3
MCNN [115]	132.7	145.3	367.6
SwitchCNN [83]	61.2	72.9	163.2
OURS-NoGeom	36.8	39.9	115.7
OURS-GeomOnly	26.1	35.3	107.2
OURS-Geom-Phy (frame interval 1)	24.8	32.7	103.2
OURS-Geom-Phy (frame interval 5)	**18.2**	**26.6**	98.7
OURS-Geom-Phy (frame interval 10)	22.9	34.3	**94.2**

Table 4.3 – Comparative results in terms of head-plane crowd density on the **Venice** dataset.

Model	MAE	$RMSE$	$MPAE$
CSRNet [45]	39.2	44.0	124.7
MCNN [115]	133.7	148.4	368.2
SwitchCNN [83]	63.1	75.8	165.4
OURS-NoGeom	37.2	40.4	116.3
OURS-GeomOnly	27.3	37.2	108.9
OURS-Geom-Phy (frame interval 1)	25.2	33.4	104.7
OURS-Geom-Phy (frame interval 5)	**18.7**	**27.0**	99.2
OURS-Geom-Phy (frame interval 10)	23.6	35.2	**95.1**

Table 4.4 – Comparative results in terms of image plane crowd density on the **Venice** dataset.

people count ranges from 54 to 301 in this dataset. We will refer to it as **Campus**.

To demonstrate that our approach also works in a very different context, we also evaluate it on the publicly available **Venice** [57] dataset, which was recorded using a mobile phone. It features Piazza San Marco as seen from various viewpoints on the second floor of the basilica and substantial perspective effects. This dataset comprises 4 different sequences and 167 annotated frames. Fig. 4.1 depicts one of these. The white lines on the Piazza make it easy to estimate the plane homography using standard photogrammetric techniques and the sequence is thus a good proxy for drone-acquired footage.

We focus on head-plane and ground-plane densities, as opposed to image-plane densities, because they are the ones that have a true physical meaning independently of the camera motion. In this section, we therefore report our results and baselines ones in head-plane density terms. However, we also provides image plane density results to demonstrate that our model outperforms the baselines in both cases.

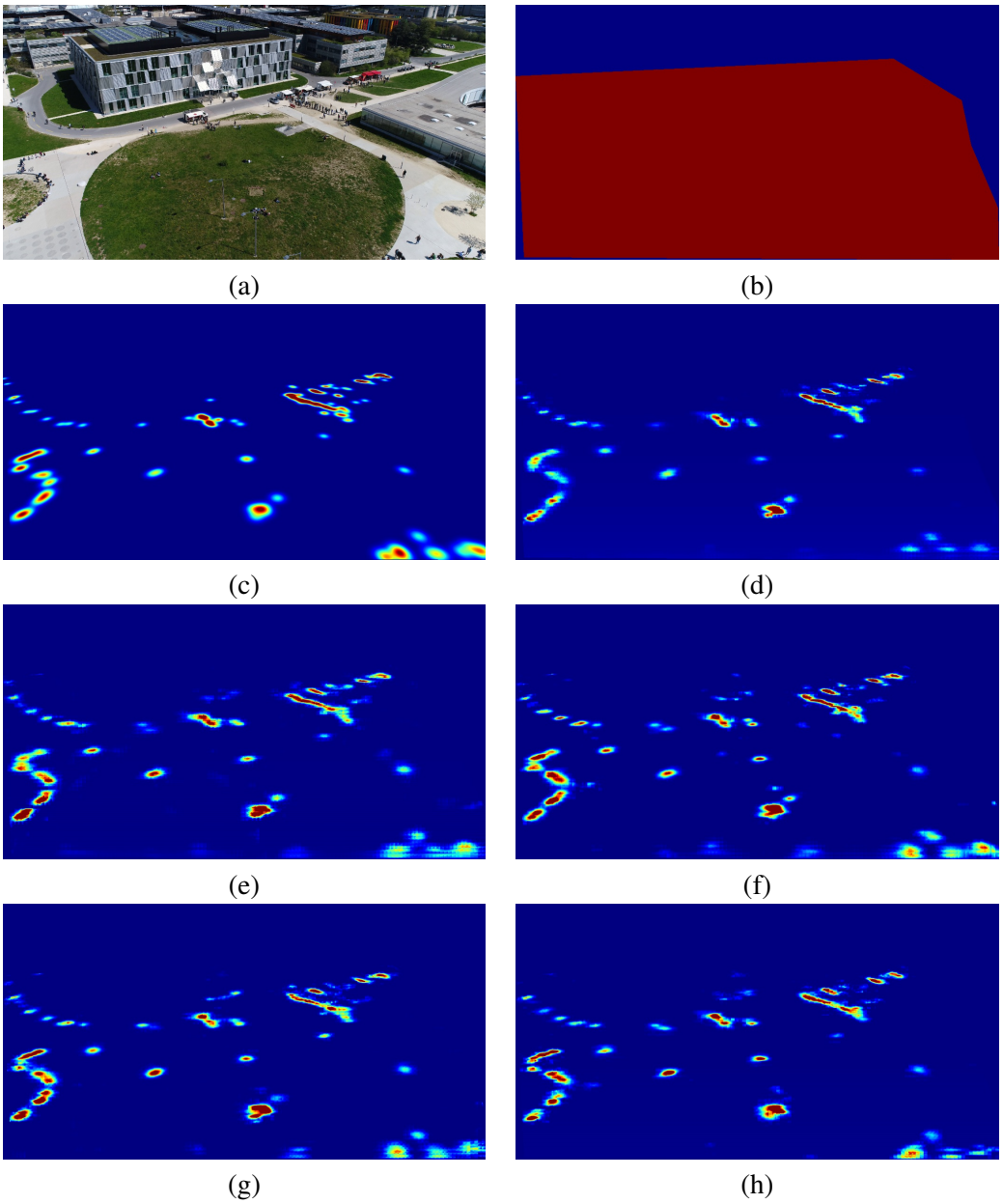


Figure 4.3 – **Crowd density estimation on the Campus dataset.** (a) Input image. (b) *ROI* overlaid in red. (c) Ground truth head plane density. (d-h) Density maps generated by **OURS-NoGeom**, **OURS-GeomOnly**, **OURS-Geom-Phy**(1), **OURS-Geom-Phy**(5), and **OURS-Geom-Phy**(10).

4.3.2 Baselines

We benchmark our approach against three recent methods for which the code is publicly available: **CSRNet** [45], **MCNN** [115] and **SwitchCNN** [83]. As discussed in the related work section, they are representative of current approaches to handling the fact that people's sizes vary depending on their distance to the camera.

We will refer to our complete approach as **OURS-Geom-Phy**. To tease out the individual contributions of its components, we also evaluate two degraded versions of it. **OURS-NoGeom** uses the CNN to predict densities but does not feed it the perspective map as input. **OURS-GeomOnly** uses the full approach described in Section 4.1 but does not impose temporal consistency.

4.3.3 Evaluation Metrics

Most previous works in crowd density estimation use mean absolute error (MAE) and root mean squared error (RMSE) as their evaluation metric. They are defined as

$$\text{MAE} = \frac{1}{N}\sum_{1}^{N}|z_i - \hat{z}_i| \text{ and RMSE} = \sqrt{\frac{1}{N}\sum_{1}^{N}(z_i - \hat{z}_i)^2}, \tag{4.14}$$

where N is the number of test images, z_i denotes the true number of people inside the ROI of the ith image and $\hat{z}_i$ the estimated number of people. While indicative, these two metrics are very coarse, since these two metrics only take into consideration the total number of people irrespective of where in the scene they may be, so they are incapable of evaluating the correctness of the spatial distribution of crowd density. A false positive in one region, coupled with a false negative in another, can still yield a perfect total number of people.

We therefore introduce one additional metric that provide finer grained measures, accounting for localization errors. We name it the mean pixel-level absolute error (MPAE) and take it to be

$$\text{MPAE} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{H}\sum_{k=1}^{W}|D_{i,j,k} - \hat{D}_{i,j,k}| \times \mathbf{1}_{\{D_{i,j,k} \in R_i\}}}{N}, \tag{4.15}$$

where $D_{i,j,k}$ is the ground-truth density of the ith image at pixel (j, k), $\hat{D}_{i,j,k}$ is the corresponding estimated density, R_i is the ROI of the ith image, $\mathbf{1}_{\{.\}}$ is the indicator function, and W and H are the image dimensions. MPAE quantifies how wrongly localized the densities are.

The baseline models [115, 45, 83] are designed to predict density in the image plane instead of the head plane, as our model does. Fortunately, the densities in image plane and head plane can be easily converted into each other, as shown in Section 4.1. For a fair comparison, we therefore train the baseline models [115, 45, 83] as reported in original the papers to estimate density in the image-plane. We then used Eq. 4.7 to convert to head-plane density. Thus we can use the MAE, RMSE, and MPAE metrics to compare both kinds of densities.

4.3.4 Quantitative Evaluation

We report our comparative results in Tables 4.1, 4.2, 4.3 and 4.4. Enforcing temporal consistency requires the central frame to be annotated but the other two can be chosen arbitrarily. When running **OURS**, that is, enforcing both geometry and temporal constraints, we used triplets of images temporally separated by 1, 5, or 10 frames. We provide a qualitative comparison in

Fig. 4.3.

In Tables 4.1 and 4.3, we used Eq. 4.7 to convert the image plane densities computed by the baselines into head-plane densities that can be compared to ours. In Tables 4.2 and 4.4, we instead converted our head plane densities into image plane ones that can be compared to theirs. Either way, **OURS-GeomOnly** outperforms the baselines. Furthermore, imposing temporal consistency gives our approach a further boost.

5 Crowd Counting with Surveillance Videos

When video sequences are available, some algorithms use temporal consistency to impose weak constraints on successive density estimates. One way is to use an LSTM to model the evolution of people densities from one frame to the next [106]. However, this does not explicitly enforce the fact that people numbers must be strictly conserved as they move about, except at very specific locations where they can move in or out of the field of view. Modeling this was attempted in the previous chapter but, because expressing this constraint in terms of people densities is difficult, the constraints actually enforced were much weaker.

In this chapter, we propose to regress people flows, that is, the number of people moving from one location to another in the image plane, instead of densities. To this end, we partition the image into a number of grid locations and, for each one, we define ten potential flows, one towards each neighboring location, one towards the location *itself*, and the last towards regions outside the image plane. The flow towards the location itself enables us to account for people who stay in the same location from one instant to the next and the final flow to account for people who enter or exit the field of view. In our experiments, we only use it at the boundaries of the image plane because there are no occluded regions in our datasets. However, if there were occluded regions within the scene, we could simply also use that last channel for motions in and out of those. In this scenario, the places where the tenth channel is to be used would have to be scene-specific and our approach offers the required flexibility. Fig. 5.1 depicts some of the ten flows we compute. All the flows incident on a grid location are summed to yield an estimate of the people density in that location. The network can therefore be trained given ground-truth estimates only of the local people densities as opposed to people flows. In other words, even though we compute flows, our network only requires ground-truth density data for training purposes, like most others.

Our formulation allows us to effectively impose people conservation constraints—people do not teleport from one region of the image to another—much more effectively than earlier approaches. This increases performance using network architectures that are neither deeper nor more complex than state-of-the-art ones. Furthermore, regressing people flows instead of densities provides a scene description that includes the motion direction and magnitude, both of which are useful

T	number of time steps
K	number of locations in the image plane
I^t	image at t-th frame
m_j^t	number of people present at location j at time t
$f_{i,j}^{t-1,t}$	number of people moving from location i to location j between times $t-1$ and t
$N(j)$	neighborhood of location j that can be reached within a single time step

Table 5.1 – **Notations.**

for crowd analytics. This also enables us to exploit the fact that people flow and optical flow should be highly correlated, as illustrated by Fig. 5.1, which provides an additional regularization constraint on the predicted flows and further enhances performance. We will demonstrate on five benchmark datasets that our approach to enforcing temporal consistency brings a substantial performance boost compared to previous approaches. We will also show that when the cameras can be calibrated, we can apply our approach in the ground plane instead of the image plane, which further improves performance.

Another key strength of our flow-based approach is that we can use it to recast our fully-supervised approach, as described above, in an Active Learning (AL) context that drastically reduces the supervision requirements without giving up accuracy. More specifically, our network learns to enforce the people conservation as best it can but they can still be violated. Our AL approach therefore involves first annotating a fraction of the training images, using them to train the network, running it on the others, selecting the areas where the constraints are most violated for further human annotation, and iterating. In effect, we use people conservation constraints to provide self-supervision and to make active learning possible. We will show that, by the time we have annotated about 6.25% of the images, we achieve almost the same accuracy as when annotating all of them and outperform previous approaches trained using full supervision.

Our contribution is therefore a novel flow-based approach to estimating people densities from video sequences that enforces strong temporal consistency constraints without requiring complex network architectures. Not only does it boost performance, it also makes it possible to implement an active-learning approach that leverages the expected consistency to reduce sixteen-fold the required amount of annotated data while preserving accuracy.

5.1 People Flows

We regress *people flows* from images. We take these flows to be counts between two consecutive time instants of people either moving from their current location to a neighboring one, staying at the same location, or moving in or out of the field of view. They are depicted by Fig. 5.2 and summarized in Table 5.1. People flows incident on a specific location are then summed to derive

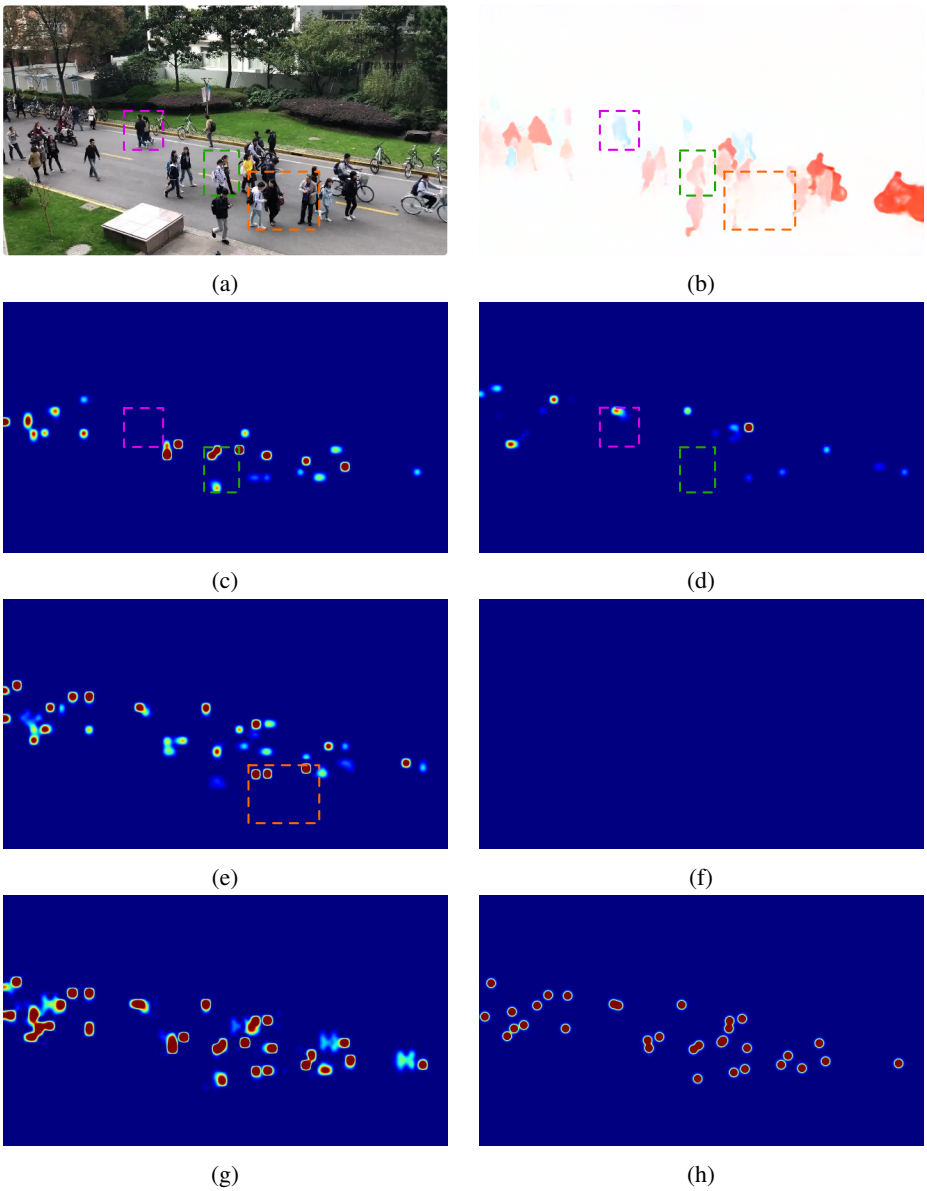
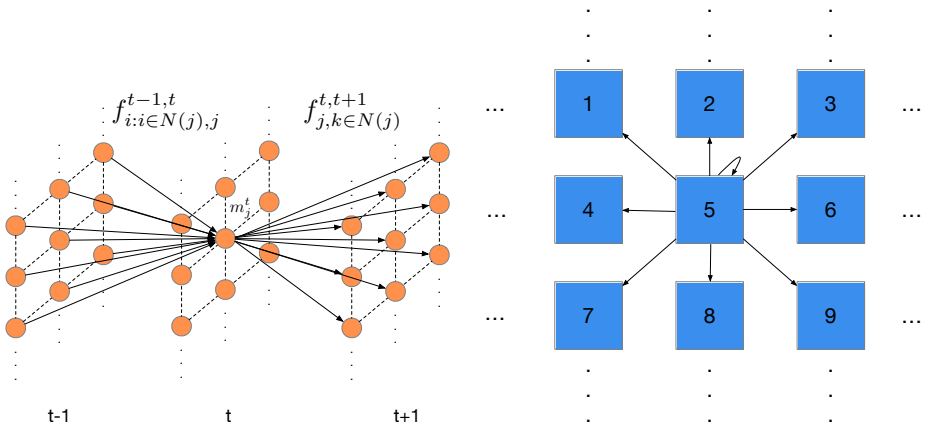


Figure 5.1 – **From people flow to crowd density.** (a) Original image. (b) Optical flow. Red denotes people moving right and blue moving left. The overlaid orange box encloses people moving slowly or not at all, the pink box people moving left, and the green box people moving right. (c) Estimated flow of people moving right. People moving left, such as those in the pink box, do not contribute to it, whereas those in the green box do. (d) Flow of people moving left. The situations within the pink and green box are reversed. (e) Estimated flow of people staying within the same grid location from one time instant to the next, such as those within the orange box. They are not necessarily static. They may simply not have had time to change location between the two time instants. (f) Estimated flow of people moving up. As no one does, it is almost zero everywhere. (g) Density map inferred by summing all the flows incident on a particular location. (h) Ground truth density map.

the number of people per location or *people count* per location. The *crowd density* then simply is the *people count* divided by the location area. Our key insight is that this formulation enables us to impose much tighter *people conservation constraints* than earlier approaches. By this, we mean that we can accurately model the fact that all people present in a location at a given instant either were already there at the previous one or came from a neighboring location. This assumes the image frequency to be high enough for people not being able to move beyond neighboring locations in the time that separates consecutive frames. This is a common assumption that has proved both valid and effective in many earlier works.

5.1.1 Formalization



(a) Grid model (b) Neighborhood of each location

Figure 5.2 – **People flows.** (a) The crowd density at time t at a given location can only come from neighboring grid locations at time $t-1$ and flow to neighboring grid locations at time $t+1$, in both cases including the location itself. (b) For each location not at the boundary of the image plane, there are nine locations reachable within a single time step, including the location itself. For locations at the edge of the image plane, we add a tenth location that represents the rest of the world. It allows for flows of people who either leave the image or enter it from outside.

Let us consider a video sequence $\mathbf{I} = \{\mathbf{I}^1, \dots \mathbf{I}^T\}$ and three consecutive images $\mathbf{I}^{t-1}$, $\mathbf{I}^t$, and $\mathbf{I}^{t+1}$ from it. Let us assume that each image has been partitioned into K rectangular grid locations. In our implementation, a location is one spatial position in the final convolutional feature map, corresponding to an 8×8 neighborhood in the image. However, other choices are possible.

The main constraint we want to enforce is that the number of people present at location j at time t is the number of people who were already there at time $t-1$ and stayed there plus the number of those who walked in from neighboring locations between $t-1$ and t. The number of people present at location j at time t also equals the sum of the number of people who stayed there until time $t+1$ and of people who went to a neighboring location between t and $t+1$.

Let m^t_j be the number of people present at location j at time t, or *people count* at that location. Let $f^{t-1,t}_{i,j}$ be the number of people who move from location i to location j between times $t-1$

and t, and $N(j)$ the neighborhood of location j that can be reached within a single time step. These notations are illustrated by Fig. 5.2 (a) and summarized in Table 5.1. In practice, we take $N(j)$ to be the 8 neighbors of grid location j plus the grid location itself to account for people who remain at the same place, as depicted by Fig. 5.2 (b). Our people conservation constraint can now be written as

$$\sum_{i \in N(j)} f_{i,j}^{t-1,t} = m_j^t = \sum_{k \in N(j)} f_{j,k}^{t,t+1} \ . \tag{5.1}$$

for all locations j that are *not* on the edge of the grid, that is, locations from which people cannot appear or disappear without being seen elsewhere in the image.

Most earlier approaches [70, 115, 16, 46, 50, 57, 54] regress the values of m_j^t, which makes it hard to impose the constraints of Eq. 5.1 because many different values of the flows $f_{i,j}^{t-1,t}$ can produce the same m_j^t values. For example, in the previous chapter, the equivalent constraint is

$$\forall j \quad m_j^t \leq \sum_{i \in N(j)} m_i^{t-1} \text{ and } m_j^t \leq \sum_{k \in N(j)} m_k^{t+1} \ . \tag{5.2}$$

It only states that the number of people at location j at time t is less than or equal to the total number of people at neighboring locations at time $t-1$ and that the same holds between times t and $t+1$. These are much looser constraints than the ones of Eq. 5.1. They guarantee that people cannot suddenly appear but do not account for the fact that people cannot suddenly disappear either. Our formulation lets us remedy this shortcoming. By regressing the $f_{i,j}^{t-1,t}$ from pairs consecutive images and computing the values of the m_j^t from these, we can impose the tighter constraints of Eq. 5.1.

5.1.2 Regressing the Flows

We now turn to the task of training a regressor that predicts flows that correspond to what is observed while obeying the above constraints and properly handling the boundary grid locations. Let us denote the regressor that predicts the flows from $\mathbf{I}^{t-1}$ and $\mathbf{I}^t$ as $\mathcal{F}$ with parameters Θ to be learned during training. In other words, $f^{t-1,t} = \mathcal{F}(I^{t-1}, I^t; \Theta)$ is the vector of predicted flows between all pairs of neighboring locations between times $t-1$ and t. In practice, $\mathcal{F}$ is implemented by a deep network. The predicted local people counts m_j^t, that is, number of people per grid location j and at time t, are taken to be the sum of the incoming flows according to Eq. 5.1, and the predicted count for the whole image is the sum of all the m_j^t. As the flows are not directly observable, the training data comes in the form of *people counts* $\bar{m}_j^t$ per grid location j and at time t.

During training, our goal is therefore to find values of Θ such that

$$\bar{m}_j^t = \sum_{i \in N(j)} f_{i,j}^{t-1,t} = \sum_{k \in N(j)} f_{j,k}^{t,t+1} \text{ and } f_{i,j}^{t-1,t} = f_{j,i}^{t,t-1} \tag{5.3}$$

for all i, j, and t, except for locations at the edges of the image plane, where people can appear from and disappear to unseen parts of the scene.

The first constraint is the people conservation constraint introduced in Section 5.1.1. The second accounts for the fact that, were we to play the video sequence in reverse, the flows should have the same magnitude but the opposite direction. As will be discussed below, we enforce these constraints by incorporating them into the loss function we minimize to learn Θ. Finally, we impose that all the flows be non-negative by using ReLU activations in the network that implements $\mathcal{F}$. Note that we only require the people flows to be non-negative; the fact that a location may contain less than 1 person simply means that the flow value will be less than 1.

Regressor Architecture. Recall that $f^{t-1,t} = \mathcal{F}(\mathbf{I}^{t-1}, \mathbf{I}^t; \Theta)$ is a vector of predicted flows from neighboring locations between times $t-1$ and t. In practice, $\mathcal{F}$ is implemented by the encoding/decoding architecture shown in Fig. 5.3, and $f^{t-1,t}$ has the same dimension as the image grid and 10 channels per location. The first are the flows to the 9 possible neighbors depicted by Fig. 5.2 (b) and the tenth represents potential flows from outside the image and is therefore only meaningful at the edges. The fifth channel denotes the flow towards the location itself, which enables us to account for people who stay in the same location from one instant to the next.

To compute $f^{t-1,t}$, consecutive frames $\mathbf{I}^{t-1}$ and $\mathbf{I}^t$ are fed to the CAN encoder network of [57]. This yields deep features $s^{t-1} = \mathcal{E}_e(I^{t-1}; \Theta_e)$ and $s^t = \mathcal{E}_e(I^t; \Theta_e)$, where $\mathcal{E}_e$ denotes the encoder with weights Θ_e. These features are then concatenated and fed to a decoder network to output $f^{t-1,t} = \mathcal{D}(s^{t-1}, s^t; \Theta_d)$, where $\mathcal{D}$ is the decoder with weights Θ_d. $\mathcal{D}$ comprises the back-end decoder of CAN [57] with an additional final ReLU layer to guarantee that the output is always non-negative.

Grid Size. In all our experiments, we treated each spatial location in the output people flow map as a separate location. Since our CAN [57] backbone outputs a down-sampled density map, each output grid location represents an 8×8 pixel block in the input image. This down-sampling rate is common in crowd counting models [57, 56, 46] because it represents a good compromise between high-resolution of the density map and efficiency of the model.

Loss Function and Training. To obtain the ground-truth maps $\bar{m}^t$ of Eq. 5.3, we use the same approach as in most previous work [70, 115, 16, 46, 50, 57, 54]. In each image $\mathbf{I}^t$, we annotate a set of s^t 2D points $P^t = \{P_i^t\}_{1 \leq i \leq s^t}$ that denote the positions of the human heads in the scene. The corresponding ground-truth density map $\bar{m}^t$ is obtained by convolving an image containing ones at these locations and zeroes elsewhere with a Gaussian kernel $\mathcal{N}(\cdot|\mu, \sigma^2)$ with mean μ and standard deviation σ. We write

$$\bar{m}_j^t = \sum_{i=1}^{s^t} \mathcal{N}(p_j|\mu = P_i^t, \sigma^2) \ , \ \forall j \ , \tag{5.4}$$

where p_j denotes the center of location j. Note that this formulation preserves the constraints of

Eq. 5.3 because we perform the same convolution across the whole image. In other words, if a person moves in a given direction by n pixels, the corresponding contribution to the density map will shift in the same direction and also by n pixels.

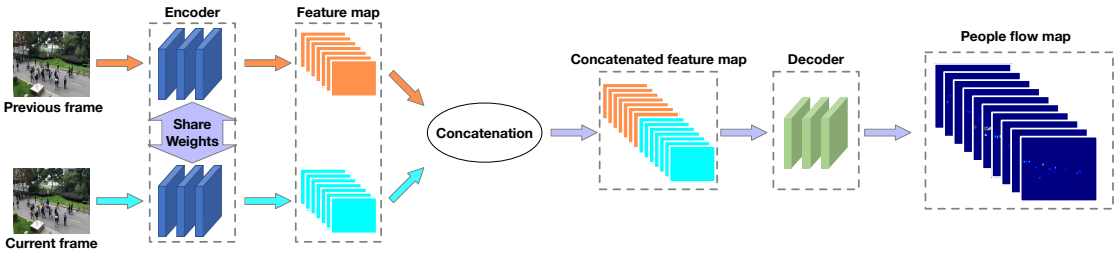


Figure 5.3 – **Model architecture:** Two consecutive RGB image frames are fed to the same encoder network that relies on the CAN scale-aware feature extractor of [57]. These multi-scale features are further concatenated and fed to a decoder network to produce the final people flow maps.

The final ReLU layer of the regressor guarantees that the estimated flows are non-negative. To enforce the constraints of Eq. 5.3, we take our combined loss function L_{combi} to be the weighted sum of two loss terms. We write

$$L_{combi} = \sum_t L^t_{flow} + \alpha L^t_{cycle} \ , \tag{5.5}$$

$$L^t_{flow} = \sum_{j \in I^t} \left[(\bar{m}^t_j - \sum_{i \in N(j)} f^{t-1,t}_{i,j})^2 + (\bar{m}^t_j - \sum_{k \in N(j)} f^{t,t+1}_{j,k})^2 \right] ,$$

$$L^t_{cycle} = \sum_{j \in I^t} \left[\sum_{i \in N(j)} (f^{t-1,t}_{i,j} - f^{t,t-1}_{j,i})^2 + \sum_{k \in N(j)} (f^{t,t+1}_{j,k} - f^{t+1,t}_{k,j})^2 \right] ,$$

where $\bar{m}^t_j$ is the ground-truth crowd density value, that is, the *people count* at time t and location j of Eq. 5.4 and α is a scalar weight we set to 1 in all our experiments.

At training time, we systematically use three consecutive frames to evaluate L_{combi} and our flow formulation requires a density map at consecutive triplet frames. A limitation of this formulation is that requires all frames to be annotated. In practice, this is not necessarily the case. In some of the examples we present in the results section, only one in 60 or 255 frames is annotated. Hence, let $\mathcal{A}$ be the set of frames that are annotated and $\mathcal{U}$ the set of their previous and next frames that are not and for which $\bar{m}^t$ is therefore unavailable. For these frames, it still holds that $\sum_{i \in N(j)} f^{t-1,t}_{i,j} = \sum_{k \in N(j)} f^{t,t+1}_{j,k}$ for all j, even if the value of the sum is unknown. We therefore rewrite our loss function as

$$L_{combi} = \sum_{t \in \mathcal{A}} L^t_{flow} + \sum_{t \in \mathcal{U}} L^t_{uflow} + \alpha \sum_t L^t_{cycle} \ , \tag{5.6}$$

$$L^t_{uflow} = \sum_{j \in I^t} (\sum_{i \in N(j)} f^{t-1,t}_{i,j} - \sum_{k \in N(j)} f^{t,t+1}_{j,k})^2 \ ,$$

where L_{flow} and L_{cycle} are defined as in Eq. 5.5. Algorithm 3 describes our training scheme in

more detail. In the results section, we show that our algorithm can handle having only one in 255 frames annotated.

Algorithm 1 Active Patch Selection Algorithm

Require: U unlabeled keyframes
Require: Pre-trained regressor network $\mathcal{F}$ using $[0.25U]$ keyframes
Require: Remaining unlabeled keyframes $\{\mathbf{I}^{o1}, \ldots, \mathbf{I}^{o[0.75U]}\}$

procedure ANNOTATION($\{\mathbf{I}^{o1}, .., \mathbf{I}^{o[0.75U]}\}$)
  for # of selection iterations **do**
    for # of unlabeled keyframes **do**
      Pick 3 consecutive frames $(\mathbf{I}^{t-1}, \mathbf{I}^{t}, \mathbf{I}^{t+1})$, where t is a multiple of V (i.e., $\mathbf{I}^{t}$ is a keyframe)
      for # of patches **do**
        Pick the l-th patch
        Compute the measure E of Eq. 5.10
      end for
      Take the maximum value E over all the patches in each unlabeled keyframe as the error for this keyframe
    end for
    Select $0.15U$ unlabeled keyframes with largest error
    For one, annotate the patch with highest value E
    Update the set of unlabeled keyframes
    Re-train $\mathcal{F}$ with all the labeled keyframes
  end for
end procedure

5.1.3 Exploiting Optical Flow

When the camera is static, both the people flow discussed above and the optical flow that can be computed directly from the images stem for the motion of the people. They should therefore be highly correlated. In fact, this remains true even if the camera moves because its motion creates an apparent flow of people from one image location to another. However, there is no simple linear relationship between people flow and optical flow. To account for their correlation, we therefore introduce an additional loss function, which we define as

$$L_{optical} = \sum_{j} \delta(m_j > 0)(\mathbf{O}_j - \bar{o}_j^{t-1,t})^2 \ , \tag{5.7}$$
$$\text{where} \quad \mathbf{O} = \mathcal{F}_o(m^{t-1}, m^t; \Theta_o) \ ,$$

m^{t-1} and m^t are density maps inferred from our predicted flows using Eq. 5.1, $\mathbf{O}_j$ denotes the corresponding predicted optical flow at grid location j by a pre-trained regressor $\mathcal{F}_o$, $\bar{o}^{t-1,t}$ is the optical flow from frames $t-1$ to t computed by a state-of-the-art optical flow network [95], and the indicator function $\delta(m_j > 0)$ ensures that the correlation is only enforced where there are

Algorithm 2 Training with Patch Annotation

Require: Training image sequence $\{\mathbf{I}^1, \ldots, \mathbf{I}^T\}$ with an interval V between annotated frames.
Require: Ground-truth density maps for one patch in every V images $\{\bar{m}_{l1}^V, \bar{m}_{l2}^{2V} ..., \bar{m}_{l(T//V)}^{(T//V)V}\}$.

procedure TRAIN($\{\mathbf{I}^1, .., \mathbf{I}^T\}$,$\{\bar{m}_{l1}^V, .., \bar{m}_{l(T//V)}^{(T//V)V}\}$)
Initialize the weights Θ of regressor network $\mathcal{F}$
for # of gradient iterations **do**
Pick 3 consecutive frames $(\mathbf{I}^{t-1}, \mathbf{I}^t, \mathbf{I}^{t+1})$, where t is a multiple of V. Only the jth patch of $\mathbf{I}^t$ is annotated
Reconstruct density map m_j^t using $\mathcal{F}(I_j^{t-1}, I_j^t, \Theta)$, $\mathcal{F}(I_j^t, I_j^{t+1}, \Theta)$, $\mathcal{F}(I_j^t, I_j^{t-1}, \Theta)$ and $\mathcal{F}(I_j^{t+1}, I_j^t, \Theta)$
Randomly select a patch q from $(\mathbf{I}^{t-1}, \mathbf{I}^t, \mathbf{I}^{t+1})$
Reconstruct density map m_q^t using $\mathcal{F}(I_q^{t-1}, I_q^t, \Theta)$, $\mathcal{F}(I_q^t, I_q^{t+1}, \Theta)$, $\mathcal{F}(I_q^t, I_q^{t-1}, \Theta)$ and $\mathcal{F}(I_q^{t+1}, I_q^t, \Theta)$
Update Θ_d using L_{advers} in Eq. 5.14 with RMSProp as suggested by [3]
Randomly select a super-patch S_k^t composed of patches from $\mathbf{I}_j^t$
Reconstruct density map of S_k^t and other unlabeled patches inside this super-patch by passing these patches through the regressor network $\mathcal{F}$
Update Θ using $L_{overall}$ in Eq. 5.11 with Adam
end for
end procedure

people. This is especially useful when the camera moves to discount the optical flows generated by the changing background. We also use CAN [57] as the optical flow regressor $\mathcal{F}_o$ with 2 input channels, one for m^{t-1} and the other m^t. This network is pre-trained separately on the training data and then used to train the people flow regressor.

Pre-training the regressor $\mathcal{F}_o$ requires annotations for consecutive frames, that is, $V = 1$ in the definition of Algorithm 3. When such annotations are available, we use this algorithm again but replace L_{combi} by

$$L_{all} = L_{combi} + \beta L_{optical} \ . \tag{5.8}$$

In all our experiments, we set β to 0.0001 to account for the fact that the optical flow values are around 4,000 times larger than the people flow values. $\mathcal{F}_o$ is also pre-trained with Adam and a learning rate of $1e-4$. During pre-training, $\mathcal{F}_o$ maps the ground-truth density map pairs $\bar{m}^{t-1}$, $\bar{m}^t$ to the optical flow map $\bar{o}^{t-1,t}$ from frames $t-1$ to t as

$$\bar{o}^{t-1,t} = \sum_j \delta(\bar{m}_j > 0) \mathcal{F}_o(\bar{m}^{t-1}, \bar{m}^t; \Theta_o) \ . \tag{5.9}$$

This pre-trained network $\mathcal{F}_o$ is then used as a regularization term when training our people flow model, using Eq. 5.7 and Eq. 5.8, where m^{t-1} and m^t are density maps obtained by summing our predicted flows.

Algorithm 3 Three-Frames Training Algorithm

Require: Training image sequence $\{\mathbf{I}^1, \ldots, \mathbf{I}^T\}$ with an interval V between keyframes.
Require: Ground-truth density maps $\{\bar{m}^V, \bar{m}^{2V} ..., \bar{m}^{(T//V)V}\}$ computed by convolving the annotations according to Eq. 5.4.

procedure TRAIN($\{\mathbf{I}^1, .., \mathbf{I}^T\}$,$\{\bar{m}^V, .., \bar{m}^{(T//V)V}\}$)
  Initialize the weights Θ of regressor network $\mathcal{F}$
  for # of gradient iterations **do**
    Pick 3 consecutive frames $(\mathbf{I}^{t-1}, \mathbf{I}^t, \mathbf{I}^{t+1})$, where t is a multiple of V
    if V=1 **then**
      Minimize L_{combi} of Eq. 5.5 w.r.t. Θ using Adam
    else
      Minimize L_{combi} of Eq. 5.6 w.r.t. Θ using Adam
    end if
  end for
end procedure

5.2 Using Less Annotated Training Data

Recall from Section 5.1.2 that we annotate only a set of keyframes. In this section, we show that we do not even need to annotate them fully. It is enough to only annotate small portions of them to pre-train the network and then exploit the flow constraints to iteratively select additional patches to be annotated. We will see in the results section that this active learning strategy allows us to achieve an accuracy that is close to what we get with full supervision at a much reduced annotation cost.

5.2.1 Patch Selection

Let us split each keyframe image $\mathbf{I}^t$ into a set of $n \times n$ patches P_k^t, where k is the patch index, as shown in Fig. 5.5. Instead of annotating whole images, we can annotate a single one of these patches in a subset of the keyframes and use the three-frame Algorithm 3 to pre-train the network. Because we use relatively little training data, it is unlikely that the values of L_{flow} and L_{cycle} of Eq. 5.5 will be zero if we evaluate the network on patches that we have *not* used for training purposes, at least not without further-training. In other words, the people conservation constraints of Eq. 5.3 will be violated. To take advantage of this, we define

$$E(P_k^t) = \sum_{j \in P_k^t} | \sum_{i \in N(j)} f_{i,j}^{t-1,t} - \sum_{k \in N(j)} f_{j,k}^{t,t+1} | \,, \tag{5.10}$$

a measure of how much the people conversation constraint is violated within patch P_k^t.

We then implement the simple patch selection strategy depicted by Fig. 5.4 and detailed by Algorithm 1. In practice, we initially annotate one patch in 25% of the keyframes, and use 60%

of them for training and the remaining 40% for validation. We train our network by minimizing the loss function L_{combi} of Eq. 5.5, whose supervised component L_{flow} is only evaluated on the annotated patches. We then forward pass the remaining keyframes through our network and, within each one, annotate the patch with the larger E. We repeat this process 5 times, selecting 15% of all the initially-unannotated keyframes at each such iteration and retraining the model with the newly-annotated image patches.

Figure 5.4 – **Our active learning pipeline.** We first annotate a fraction of the training image patches, use them to train the network while minimizing the consistency and adversarial loss terms, and then run inference on the others. We then select patches where the people conservation constraints are most violated for further human annotation and iterate the process.

5.2.2 Adding New Terms to the Objective Function

In the fully supervised case, there was no need to enforce spatial consistency across patches in the *same* image because the ground-truth data did it implicitly. However, in the scenario where we have ground-truth data for only a small subset of the patches, this has to be done explicitly. Furthermore, we must avoid overfitting to the labeled patches.

To achieve these two goals, we introduce two additional loss terms $L_{spatial}$ and L_{advers} described in the remainder of this section, and thus minimize the overall loss

$$L_{overall} = L_{combi} + \gamma L_{spatial} + \delta L_{advers} \ , \tag{5.11}$$

where γ and δ are weighing factors. The training strategy is detailed by Algorithm 2.

Spatial People Conservation Loss: $L_{spatial}$

To handle the scenario where we have ground-truth data for only a subset of the patches, we replace the missing ground-truth data by spatial consistency constraints as follows. Let us consider keyframe $\mathbf{I}^t$ that has been split into patches $\{P_k^t\}$ and assume that we have annotated P_j^t only. We define S_k^t as a *super-patch* composed of P_j^t and unannotated patches P_k^t for $k \in \mathcal{P}_j$, where $\mathcal{P}_j$ is a set of at most 15 indices, randomly chosen each time we compute the spatial loss. In other words, this means that a super-patch can range between the entire image and the combination of P_j^t with a single immediate neighbor. We then pass each patch through the network individually to obtain people counts m_k^t for $k \in \mathcal{P}_j$, and further forward pass the super-patch through the network to compute people counts M_k^t. Because the number of people in

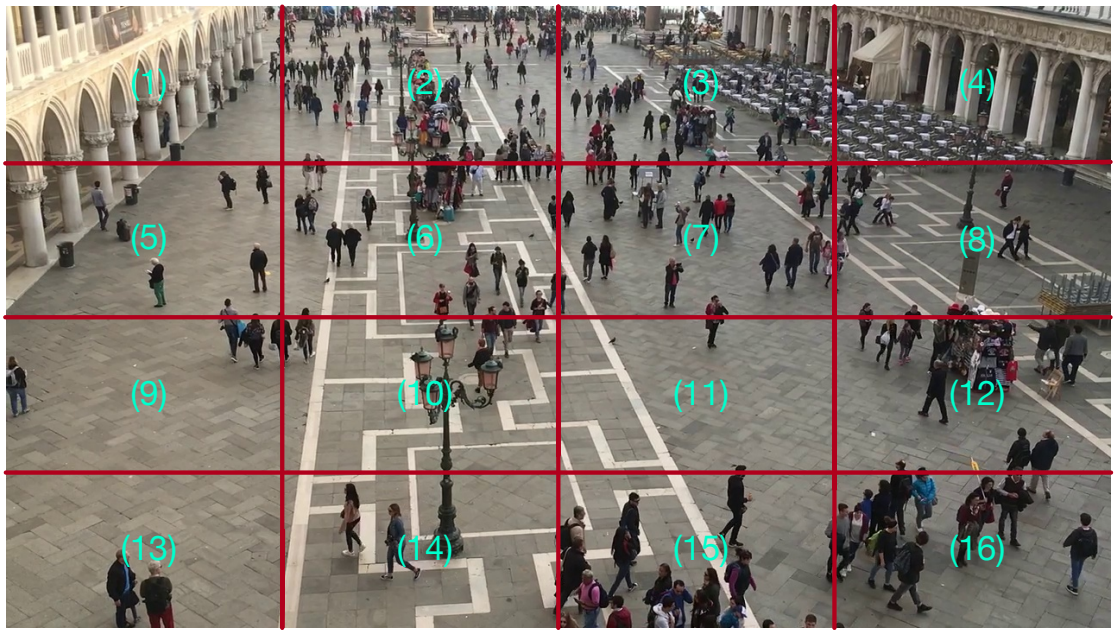


Figure 5.5 – **Spatial people conservation constraint.** An image from **Venice** [57] dataset, we could split this image into 4×4 patches. Any adjacent $a \times a$ patches would constitute a super-patch. The spatial people conservation constraint hold between any super-patch and all the patches inside it. For example, if we only annotate the 15th patch, one of the people conservation constraint is that the number of people in a super-patch that consists of the 11th, 12th, 15th and 16th patches, equals to the sum of the number of people in the 11th, 12th, 15th and 16th patches.

the super-patch must be the sum of the number of people in each individual patch, we should have

$$\forall j, \ , \sum_{i \in P_j^t} m_i^t + \sum_{k \in \mathcal{P}_j} \sum_{i \in P_k^t} m_i^t = \sum_{i \in S_j^t} M_i^t \ . \tag{5.12}$$

We therefore write

$$L_{spatial} = \sum_j (\sum_{i \in P_j^t} m_i^t + \sum_{k \in \mathcal{P}_j} \sum_{i \in P_k^t} m_i^t - \sum_{i \in S_j^t} M_i^t)^2 \ . \tag{5.13}$$

When we take the super-patch as input, the receptive field for the corresponding sub-patch is larger than the sub-patch itself. By contrast if we only take the sub-patch as input, the receptive field is limited to it. Therefore, our loss term encourages the estimated densities for unlabeled sub-patches to be consistent independently of the contextual information.

Adversarial Loss: L_{advers}

To prevent overfitting, we further introduce an adversarial loss term inspired by GANs [29]. We take the generator to be the function $\mathcal{G}$ that runs our flow-predicting network $\mathcal{F}(\cdot, \cdot, , \Theta_d)$ on a pair of images $(\mathbf{I}^{t-1}, \mathbf{I}^t)$ and infers from it a people-density in $\mathbf{I}^t$ by summing the flows according

to Eq. 5.1. We then define a discriminator $\mathcal{D}(\cdot, \Theta_d)$ as a multilayer perceptron that takes as input the people-density map $\{m_i^t, i \in P_k^t\}$ and returns the probability that it comes from a patch that has been annotated. Let A be the set of patches that have been annotated. We write

$$L_{advers} = -\sum_{P \in A} \log(\mathcal{D}(m_i^t)) - \sum_{P \notin A} \log(1 - \mathcal{D}(m_i^t)) \ . \tag{5.14}$$

5.3 Experiments

In this section, we first introduce the evaluation metrics and benchmark datasets used in our experiments. We then show that our fully supervised approach outperforms previous methods when operating in the image plane and does even better when image registration is available by working in the ground plane instead of the image plane. We then quantify the ability of our active learning algorithm to reduce the annotation cost.

5.3.1 Evaluation Metrics

Previous works in crowd density estimation use the mean absolute error (MAE) and the root mean squared error ($RMSE$) as evaluation metrics [115, 111, 70, 83, 106, 91]. They are defined as

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|z_i - \hat{z}_i| \text{ and } RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(z_i - \hat{z}_i)^2} \ .$$

where N is the number of test images, z_i denotes the true number of people inside the ROI of the ith image and $\hat{z}_i$ the estimated number of people. In the benchmark datasets discussed below, the ROI is the whole image except when explicitly stated otherwise. In practice, $\hat{z}_i$ is taken to be $\sum_{p \in I_i} m_p$, that is, the sum over all locations or people counts obtained by summing the predicted people flows.

5.3.2 Benchmark Datasets and Ground-truth Data

For evaluations purposes, we use five different datasets, for which the videos have been released along with recently published papers. The first one is a synthetic dataset with ground-truth optical flows. The other four are real world videos, with annotated people locations but without ground-truth optical flow. To use the optional optical flow constraints introduced in Section 5.1.3, we therefore use the pre-trained **PWC-Net** [95] to compute the loss function $L_{optical}$ of Eq. 5.7. Fig. 5.6 depicts one such flow.

CrowdFlow [84]. This dataset consists of five synthetic sequences ranging from 300 to 450 frames each. Each one is rendered twice, once using a static camera and the other a moving one. The ground-truth optical flow is provided as shown at Fig. 5.7. As this dataset has not been used for crowd counting before, and the training and testing sets are not clearly described in [84], to

Figure 5.6 – **Estimated optical flow in FDST.** An image and the corresponding optical flow estimated using **PWC-Net** [95].

verify the performance difference caused by using ground-truth optical flow vs. estimated one, we use the first three sequences of both the static and moving camera scenarios for training and validation, and the last two for testing.

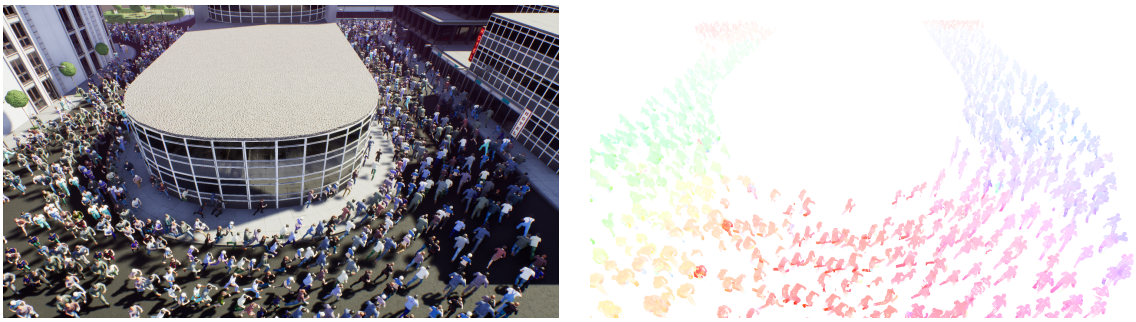


Figure 5.7 – **Ground-truth optical flow in CrowdFlow.** (left) Original image. (Right) Corresponding optical flow map.

FDST [25]. It comprises 100 videos captured from 13 different scenes with a total of 150,000 frames and 394,081 annotated heads. The training set consists of 60 videos, 9000 frames and the testing set contains the remaining 40 videos, 6000 frames. We use the same setting as in [25].

UCSD [17]. This dataset contains 2000 frames captured by surveillance cameras on the UCSD campus. The resolution of the frames is 238 $\times$ 158 pixels and the framerate is 10 fps. For each frame, the number of people varies from 11 to 46. We use the same setting as in [17], with frames 601 to 1400 used as training data and the remaining 1200 frames as testing data.

Venice [57]. It contains 4 different sequences and in total 167 annotated frames with fixed 1,280 $\times$ 720 resolution. As in [57], 80 images from a single long sequence are used as training data. The remaining 3 sequences are used for testing purposes.

WorldExpo'10 [111]. It comprises 1,132 annotated video sequences collected from 103 different scenes. There are 3,980 annotated frames, 3,380 of which are used for training purposes. Each scene contains a Region Of Interest (ROI) in which the people are counted. As in previous work [111, 115, 83, 81, 46, 16, 50, 91, 85, 74, 88] on this dataset, we report the *MAE* of each

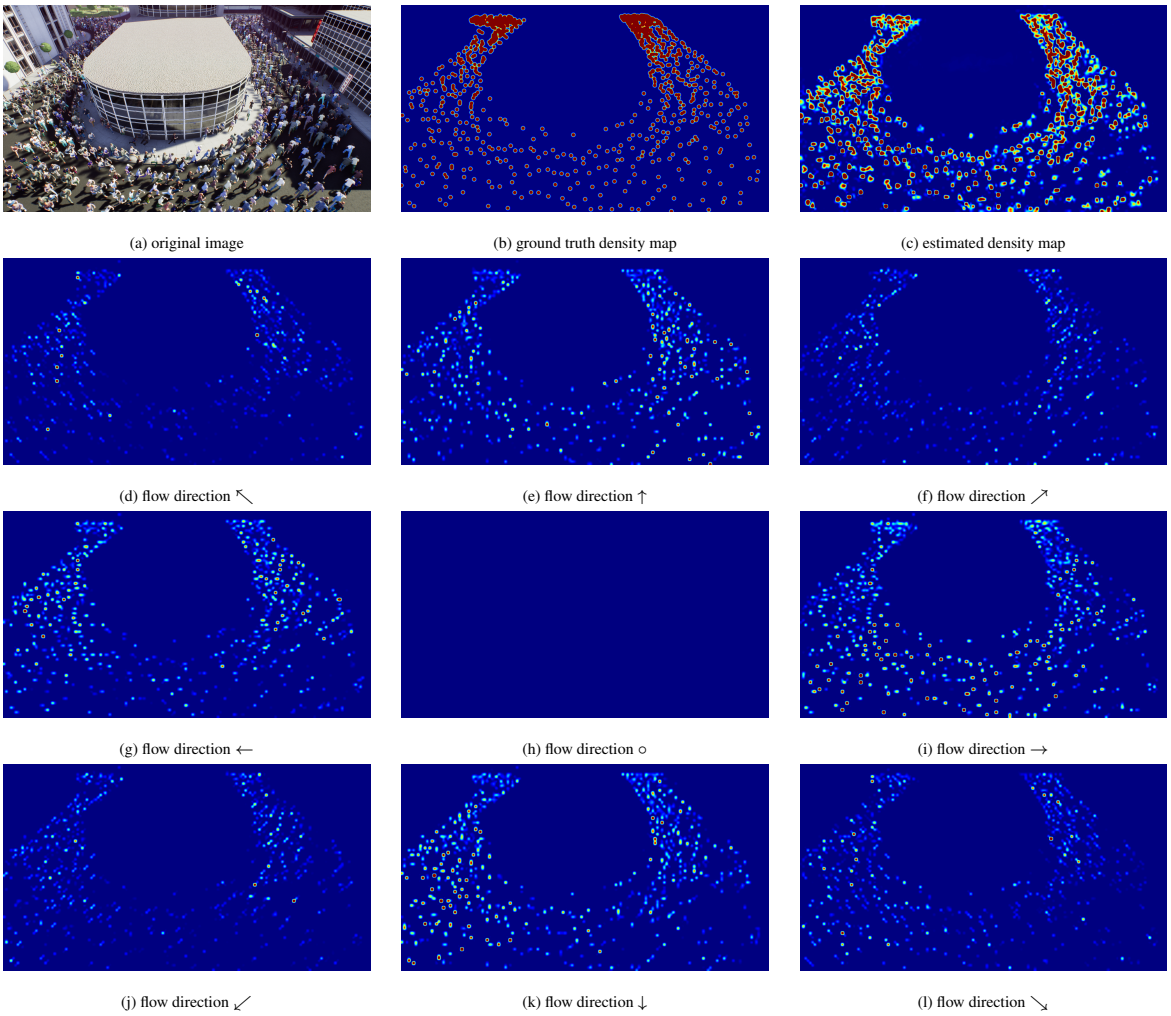


Figure 5.8 – **Density estimation in CrowdFlow.** People are running counterclockwise. The estimated people density map is close to the ground-truth one. It was obtained by summing the flows towards the 9 neighbors of Fig. 5.2 (b). They are denoted by the arrows and the circle. The latter corresponds to people not moving and is, correctly, empty. Note that the flow of people moving down is highest on the left of the building, moving right below the building, and moving up on the right of the building, which is also correct. Inevitably, there is also some noise in the estimated flow, some of which is attributable to body shaking while running.

scene, as well as the average over all scenes.

For **CrowdFlow**, **FDST** and **UCSD**, all frames in the training set are annotated. For **Venice** and **WorldExpo'10**, annotations are only available for every 60 and 255 frames, respectively.

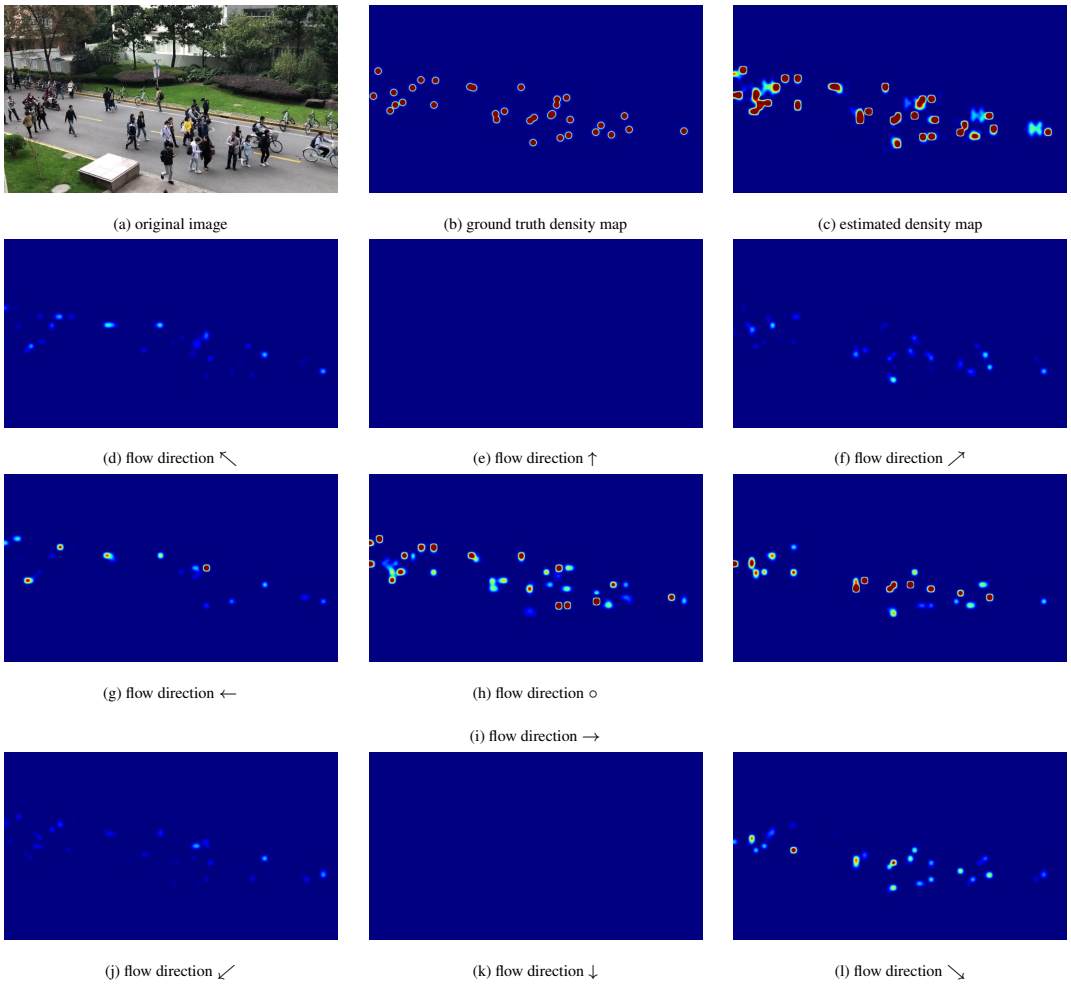


Figure 5.9 – **Density estimation in FDST.** People mostly move from left to right. The estimated people density map is close to the ground-truth one. It was obtained by summing the flows towards the 9 neighbors of Fig. 5.2 (b). They are denoted by the arrows and the circle. Strong flows occur in (g),(h), and (i), that is, moving left, moving right, or not having moved. Note that the latter does not mean that the people are static but only that they have not had time to change grid location between the two time instants.

5.3.3 Fully Supervised Approach

Comparing against Recent Techniques

We denote our model trained using the combined loss function L_{combi} of Section 5.1.2 as **OURS-COMBI** and the one using the full loss function L_{all} of Section 5.1.3 with ground-truth optical flow as **OURS-ALL-GT**. In other words, **OURS-ALL-GT** exploits the optical flow while **OURS-COMBI** does not. If the ground-truth optical flow is not available, we use the optical flow estimated by **PWC-Net** [95] and denote this model as **OURS-ALL-EST**.

Synthetic Data. Fig. 5.8 depicts a qualitative result, and we report our quantitative results on the

Model	Temporal	MAE	$RMSE$
MCNN [115]		172.8	216.0
CSRNet[46]		137.8	181.0
CAN[57]		124.3	160.2
OURS-COMBI	✓	97.8	112.1
OURS-ALL-EST	✓	96.3	111.6
OURS-ALL-GT	✓	**90.9**	**110.3**

(a)

Model	Temporal	MAE	$RMSE$
MCNN [115]		3.77	4.88
ConvLSTM [106]	✓	4.48	5.82
WithoutLST [25]		3.87	5.16
LST [25]	✓	3.35	4.45
CAN [57]		2.44	2.96
OURS-COMBI	✓	2.17	2.62
OURS-ALL-EST	✓	**2.10**	**2.46**

(b)

Model	Temporal	MAE	$RMSE$
MCNN [115]		145.4	147.3
Switch-CNN [83]		52.8	59.5
CSRNet[46]		35.8	50.0
CAN[57]		23.5	38.9
ECAN[57]		20.5	29.9
GPC[56]	✓	18.2	26.6
OURS-COMBI	✓	15.0	19.6
OURS-ALL-EST	✓	14.2	18.4
OURS-COMBI-GROUND	✓	**12.3**	**17.1**

(c)

Model	Temporal	MAE	RMSE
Zhang *et al.* [111]		1.60	3.31
Hydra-CNN [70]		1.07	1.35
CNN-Boosting [99]		1.10	-
MCNN [115]		1.07	1.35
Switch-CNN [83]		1.62	2.10
ConvLSTM [106]	✓	1.30	1.79
Bi-ConvLSTM [106]	✓	1.13	1.43
ACSCP [85]		1.04	1.35
CSRNet [46]		1.16	1.47
SANet [16]		1.02	1.29
ADCrowdNet [54]		0.98	1.25
PACNN [86]		0.89	1.18
SANet+SPANet [21]		1.00	1.28
CAN [57]		0.98	1.26
OURS-COMBI	✓	0.86	1.13
OURS-ALL-EST	✓	**0.81**	**1.07**

(d)

Model	Temporal	Scene1	Scene2	Scene3	Scene4	Scene5	**Average**
Zhang *et al.* [111]		9.8	14.1	14.3	22.2	3.7	12.9
MCNN [115]		3.4	20.6	12.9	13.0	8.1	11.6
Switch-CNN [83]		4.4	15.7	10.0	11.0	5.9	9.4
CP-CNN [91]		2.9	14.7	10.5	10.4	5.8	8.9
ACSCP [85]		2.8	14.05	9.6	8.1	2.9	7.5
IG-CNN [81]		2.6	16.1	10.15	20.2	7.6	11.3
ic-CNN[74]		17.0	12.3	9.2	8.1	4.7	10.3
D-ConvNet [88]		**1.9**	12.1	20.7	8.3	**2.6**	9.1
CSRNet [46]		2.9	11.5	8.6	16.6	3.4	8.6
SANet [16]		2.6	13.2	9.0	13.3	3.0	8.2
DecideNet [50]		2.0	13.14	8.9	17.4	4.75	9.23
CAN [57]		2.9	12.0	10.0	**7.9**	4.3	7.4
ECAN [57]		2.4	**9.4**	8.8	11.2	4.0	7.2
PGCNet [109]		2.5	12.7	8.4	13.7	3.2	8.1
OURS-COMBI	✓	2.2	10.8	**8.0**	8.8	3.2	6.6
OURS-ALL-EST	✓	2.1	10.9	8.5	8.4	3.0	**6.58**

(e)

Table 5.2 – **Comparative results on different datasets.** (a) **CrowdFlow**. (b) **FDST**. (c) **Venice**. (d) **UCSD**. (e) **WorldExpo'10**.

CrowdFlow dataset in Table 5.2 (a). **OURS-COMBI** outperforms the competing methods by a significant margin while **OURS-ALL-EST** delivers a further improvement. Using the ground-truth optical flow values in our L_{all} loss term yields yet another performance improvement, that points to the fact that using better optical flow estimation than **PWC-Net** [95] might help.

Real Data. Fig. 5.9 depicts a qualitative result, and we report our quantitative results on the four real-world datasets in Tables 5.2 (b), (c), (d) and (e). For **FDST** and **UCSD**, annotations in consecutive frames are available, which enabled us to pre-train the $\mathcal{F}_o$ regressor of Eq. 5.7. By contrast, for **Venice** and **WorldExpo'10**, only a sparse subset of frames are annotated, and we therefore warp the crowd annotation using optical flow estimation from **PWC-NET** [95]. We

report results for both **OURS-COMBI** and **OURS-ALL-EST**.

For **FDST**, **UCSD**, and **Venice**, our approach again clearly outperforms the competing methods, with the optical flow constraint further boosting performance when applicable. For **World-Expo'10**, the ranking of the methods depends on the scene being used, but ours still performs best on average and on Scene3. In short, when the crowd is dense, our approach dominates the others. By contrast, when the crowd becomes very sparse as in Scene1 and Scene5, models that comprise a pool of different regressors, such as [88], gain an advantage. This points to a potential way to further improve our own method, that is, to also use a pool of regressors to estimate the people flows.

Recall that for **FDST** and **UCSD** all training frames are annotated whereas only a fraction are for **Venice** and **WorldExpo'10**, which demonstrates that our approach can handle a large number of unannotated frames.

Working in the Ground Plane

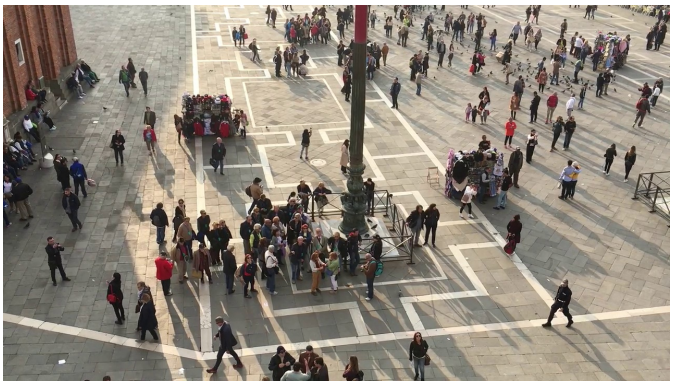
Until now, we have performed all the computations in image space, in large part so that we can compare our results to that of other recent algorithms that also work in image space. However, this neglects perspective effects as people densities per unit of image area are affected by where in the image the pixels are. To account for them, we can model them by working in the ground plane instead of the image plane, which we do in this section.

Let $\mathbf{H}^i$ be the homography from image I^i to the corresponding ground plane. We define the ground-truth density as a sum of Gaussian kernels centered on peoples' heads on the ground plane. Because we now work in the physical world, we can use the same kernel size across the entire scene and across all scenes. A head annotation P^i, that is, a 2D image point expressed in projective coordinates, is mapped to $\mathbf{H}^i P^i$ on the ground plane. Given a set $A^i = \{P^i_1, ..., P^i_{c_i}\}$ of c^i such annotations, we take the *ground plane density* G^i at point P expressed in ground plane coordinates to be

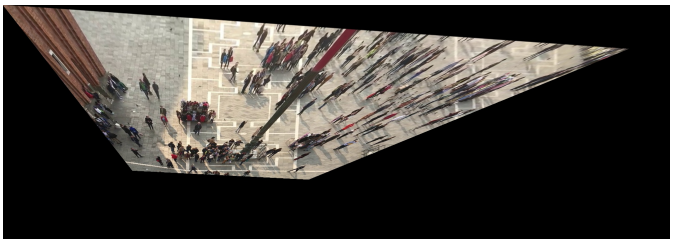
$$G^i(P) = \sum_{j=1}^{c^i} \mathcal{N}(P|\mathbf{H}^i P^i_j, \sigma) \ , \tag{5.15}$$

where $\mathcal{N}(.|\mu, \sigma)$ is a 2D Gaussian kernel with mean μ and variance σ. Note the difference compared with image plane crowd density, which is defined at Eq. 5.4. If we take our grid cells to be 30cm square and use a 30 FPS video, no one going slower than 9m/s, i.e., 32.5 km/h, can exit the neighborhood of its current location between two frames, which is more than enough for most humans. For faster animals, we would have to work with larger grid cells, more extended neighborhoods, or a higher frame rate.

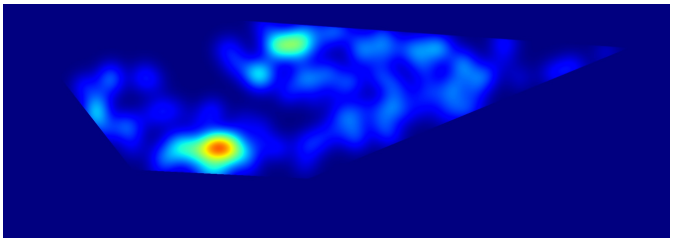
Since **Venice** is the only publicly available video-based single-view crowd counting dataset containing accurate camera pose information, it is the one we used to evaluate this approach. The ground plane regressor architecture is the same as before, with an additional Spatial Transformer



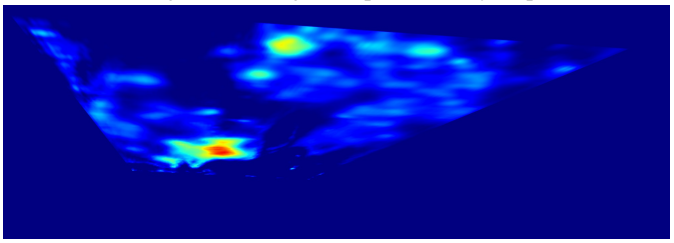
(a) image plane image



(b) ground plane image



(c) ground truth ground plane density map



(d) estimated ground plane density map

Figure 5.10 – **Ground plane density estimation in Venice.** An image and its corresponding ground plane density map estimation.

Networks [38] to map the output to the ground plane. The results are denoted by **OURS-COMBI-GROUND** in Table 5.2(c) and show a marked improvement over **OURS-COMBI** that operates strictly in the image plane. Fig. 5.10 depicts corresponding density estimates in the image and ground planes.

Ablation Study

We know examine the individual components of our fully-supervised approach and show that each one contributes to these results.

People Flows vs People Densities. To confirm that the good performance we report really is attributable to our regressing flows instead of densities, we performed the following set of experiments. Recall from Section 5.1, that we use the CAN [57] architecture to regress the flows. Instead, we can use it to directly regress the densities, as in the original CAN paper. We will refer to this approach as **BASELINE**. In the previous chapter, it was suggested that people conservation constraints could be added by incorporating a loss term that enforces the conservation constraints of Eq. 5.2 that are weaker than those of Eq. 5.1, that is, those we use in this chapter. We will refer to this approach relying on weaker constraints while still using the CAN backbone as **WEAK**. As **OURS-COMBI**, it takes two consecutive images as input. For the sake of completeness, we also implemented a simplified approach, **IMAGE-PAIR**, that takes the same two images as input and directly regresses the densities. To show that regressing flows is more effective than simply smoothing the densities, we implement **AVERAGE** that takes three images as input, uses CAN to independently compute three density maps, and then averages them. Finally, to highlight the importance of the forward-backward constraints of Eq. 5.3, we also tested a simplified version of our approach in which we drop them and that we refer to as **OURS-FLOW**.

We compare the performance of these five approaches on **CrowdFlow**, **FDST**, and **UCSD** in Table 5.3. Both **IMAGE-PAIR** and **AVERAGE** do worse than **BASELINE**, which confirms that temporal averaging of the densities is not the right thing to do. As reported in the previous chapter, **WEAK** delivers a small improvement. As expected **OURS-FLOW** improves on **IMAGE-PAIR** in all three datasets, with further performance increase for **OURS-COMBI** and **OURS-ALL-EST**. This confirms that using people flows instead of densities is a win and that the additional constraints we impose all make positive contributions.

Training the Optical Flow Regressor. As explained in Section 5.1.3, we use optical flow to regularize the people flow estimates. To this end, we need to train the regressor $\mathcal{F}_o$ of Eq. 5.7 that associates to consecutive density images an optical flow estimate that can be compared to that produced by a state-of-the-art optical flow estimator. In our implementation, $\mathcal{F}_o$ takes as input the density images but *not* the original images, our intuition being that if it did, it could predict the correct optical flows even if the density estimates were wrong, which would defeat its purpose. To confirm this, we implemented a version called **OURS-IMG-FLOW** in which $\mathcal{F}_o$ takes both the original images and crowd density maps as input. As can be seen in Table 5.4, the results are less good.

Using the Spatial Loss Term. Our active learning approach of Section 5.2 relies on the spatial loss term $L_{spatial}$ of Eq. 5.13, which we do not normally use in the fully-supervised case, essentially because minimizing it imposes constraints that are weaker than those than the flow-

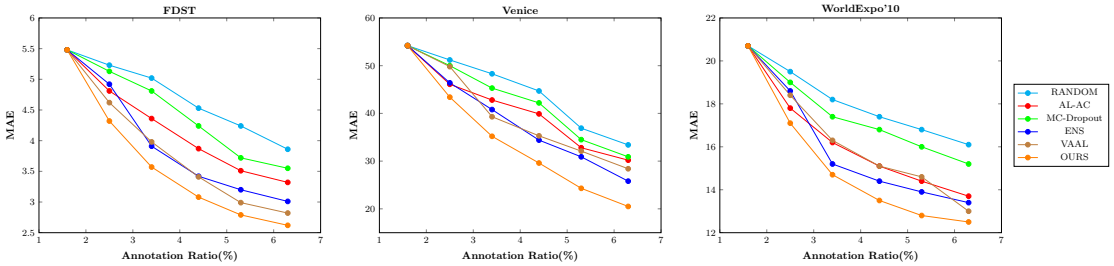


Figure 5.11 – **Comparing against other AL approaches.** We plot the MAE obtained using different active learning algorithms as a function of the annotation ratio. All models were initially trained with 25% randomly selected images of which only $1/16$ of the area was annotated. At each active learning iteration, another 15% of the images were selected either randomly or actively and another $1/16$th annotated. All the models are trained using the same loss function, the only difference being how the patches are selected. Our AL approach consistently outperform others in all the datasets.

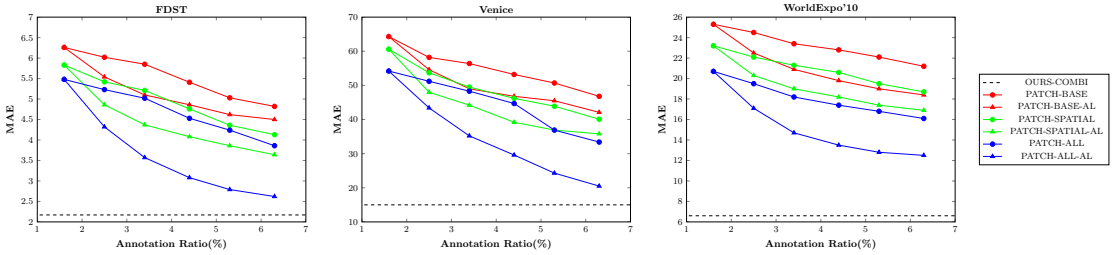


Figure 5.12 – **Ablation study of our AL approach.** We plot the MAE obtained using different versions of our AL strategy as a function of the annotation ratio. As expected, our complete approach does best.

consistency constraints of Eq. 5.3 impose. To check the validity of this choice, we implemented a variant of our approach that includes this additional loss term and that we refer to as **OURS-COMBI-SPA**. As can be in seen in Table 5.5, it performs very comparably to **OURS-COMBI**, as could be expected.

Forward vs Backward Flows. In our approach we compute both forward flows of the form $f_{i,j}^{t-1,t}$ and backward flows of the form $f_{j,i}^{t,t-1}$ and we can sum either to obtain the people densities. Let **OURS-COMBI-FOR** and **OURS-COMBI-BACK** be versions of our approach that does either, whereas **OURS-COMBI** averages the two values, which provides a slight boost as can be seen in Table 5.5.

5.3.4 Active Learning with Self-Supervision

Recall from Section 5.3.3 that **OURS-COMBI** denotes our full approach when taking a single image as input, that is, without exploiting temporal consistency. Here, we combine it with the active learning strategies for Section 5.2.

Model	People Flow	Cycle Consistency	Optical Flow	CrowdFlow MAE	CrowdFlow $RMSE$	UCSD MAE	UCSD $RMSE$	FDST MAE	FDST $RMSE$
BASELINE				124.3	160.2	0.98	1.26	2.44	2.96
IMAGE-PAIR				125.7	164.1	1.02	1.40	2.48	3.10
AVERAGE				128.9	174.6	1.01	1.31	2.52	3.14
WEAK [56]				121.2	155.7	0.96	1.30	2.42	2.91
OURS-FLOW	✓			113.3	140.3	0.94	1.21	2.31	2.85
OURS-COMBI	✓	✓		97.8	112.1	0.86	1.13	2.17	2.62
OURS-ALL-EST	✓	✓	✓	**96.3**	**111.6**	**0.81**	**1.07**	**2.10**	**2.46**

Table 5.3 – **People flow vs people densities.** The tick marks indicate what subset of the consistency constraints each method uses.

Comparing against Recent Techniques

Here, we compare our patch selection strategy against other AL approaches in the same setting.

- AL-AC [117]: It is a recent approach to active crowd counting, it actively choose the unlabeled images with high dissimilarity in crowd density distribution compared with the labeled one. Besides, a discriminator classifier is also added to distinguish if the sample is labeled or not.
- MC-Dropout [27]: It measures the uncertainty by sampling from the average output of multiple forward passes with random dropout masks. Samples with high uncertainty are selected for training in next iteration.
- ENS [8]: It is an ensemble-based approach which measures the uncertainty by sampling from the average output of multiple forward passes of different models trained with different initialization . Same as MC-Dropout, samples with high uncertainty are selected for training in next iteration.
- VAAL [94]: It learns a latent space using a variational auto encoder (VAE) and an adversarial network trained to discriminate between unlabeled and labeled data. The samples predicted to be unlabeled with high probability are chosen to annotate in next iteration.

We extend the above approaches in the same setting as ours with the same crowd density regressors. All models are trained using the same loss function $L_{overall}$ of Eq. 5.11. The only difference is how we select the patches to annotate. We evaluate the various approaches on **FDST**, **Venice** and **WorldExpo'10**. As can be seen in Fig. 5.11, our approach consistently outperforms the others.

Ablation Study

We now turn to the individual components of our active-learning scheme and implemented the following variants to gauge their impact:

Model	People Flow	Cycle Consistency	Optical Flow	CrowdFlow MAE	CrowdFlow $RMSE$	UCSD MAE	UCSD $RMSE$	FDST MAE	FDST $RMSE$
OURS-IMG-FLOW	✓	✓	✓	97.5	110.7	0.85	1.21	2.15	2.74
OURS-ALL-EST	✓	✓	✓	**96.3**	**111.6**	**0.81**	**1.07**	**2.10**	**2.46**

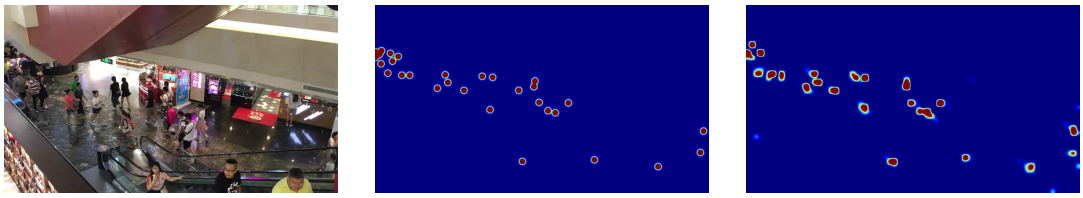
Table 5.4 – **Training the optical flow regressor**

Model	People Flow	Cycle Consistency	Optical Flow	CrowdFlow MAE	CrowdFlow $RMSE$	UCSD MAE	UCSD $RMSE$	FDST MAE	FDST $RMSE$
OURS-COMBI-FOR	✓	✓		98.0	112.6	0.87	1.19	2.19	2.65
OURS-COMBI-BACK	✓	✓		98.1	112.4	0.88	1.14	2.18	2.63
OURS-CAN	✓	✓		**97.7**	112.4	**0.86**	1.15	2.18	**2.59**
OURS-COMBI	✓	✓		97.8	**112.1**	**0.86**	**1.13**	**2.17**	2.62

Table 5.5 – **Using the spatial loss term and reversing the flows**

- **PATCH-BASE**. The model is trained using a single patch per image by only minimizing the supervised loss function L_{combi} of Eq. 5.5 and randomly selecting the patch to annotate.
- **PATCH-BASE-AL**. The model is trained using the same loss as **PATCH-BASE** but we actively select the patch to annotate using the measure of consistency violation of Eq. 5.10.
- **PATCH-SPATIAL**. The model is trained using the combined loss function including L_{combi} and $L_{spatial}$ of Eq. 5.5 and Eq. 5.13; the patch is selected randomly.
- **PATCH-SPATIAL-AL**. The model is trained using the same loss as **PATCH-SPATIAL** but we actively select the patch to annotate using the measure of consistency violation of Eq. 5.10.
- **PATCH-ALL**. The model is trained with the complete loss function $L_{overall}$ of Eq. 5.11; the patch to annotate is selected randomly.
- **PATCH-ALL-AL**. The model is trained using the same loss as **PATCH-SPATIAL** and we actively select the patch to annotate using the measure of consistency violation of Eq. 5.10.

For all models, we start by randomly selecting 25% of the training images, each of which is split into 4×4 patches, only one of which is annotated. Therefore, the starting annotation rate is $25\%/16 = 1.5625\%$. During each active learning iteration, another 15% of the training images are selected, and we also annotate one patch of each image. After 5 iterations, only 6.25% of the training patches have been selected, and we measured the ratio of annotated people to be around 5.7%. Fig. 5.12 depicts the MAE on **FDST**, **Venice** and **WorldExpo'10**. Note that both our loss terms and the AL algorithm consistently improve the performance with the largest boost coming from the active patch selection strategy. Furthermore, as can be seen by comparing these results with those in Tables 5.2 (b), (c) and (e), even though **PATCH-ALL-AL** only uses 6.25% of the annotations, it outperforms several SOTA models trained with full supervision. Fig. 5.13 depicts an example density map inferred by **PATCH-ALL-AL**.



(a) Input image (b) Ground truth crowd density map (b) Our prediction

Figure 5.13 – **Example crowd density map prediction with less annotation.** (a) Example test image from **FDST** [25] dataset (b) Ground truth crowd density map (c) Inferred crowd density map. Note how similar our prediction is to the ground truth one even though only a $1/16$ patch of each image is annotated in the training dataset.

6 Concluding Remarks

In this thesis, we have presented several solutions to crowd counting problem with different input modalities. In the following, we first briefly summarize achievements and contributions presented in this thesis and then discuss some limitations of our approaches and identify several potential directions for future research.

6.1 Summary

In Chapter 3, we target on crowd counting with random single images. We introduce a deep architecture that explicitly extracts features over multiple receptive field sizes and learns the importance of each such feature at every image location, thus accounting for potentially rapid scale changes. In other words, our approach adaptively encodes the scale of the contextual information necessary to predict crowd density. This is in contrast to crowd-counting approaches that also use contextual information to account for scaling effects as in previous work, but only in the loss function as opposed to computing true multi-scale features as we do.

In Chapter 4, we work on crowd counting with aerial videos. Apart from input video sequence, we also have detailed scene geometry information provided by drone sensors. We have shown that providing to a deep net an explicit model of perspective distortion effects as an input, along with enforcing physics-based spatio-temporal constraints, substantially increases performance. In particular, it yields not only a more accurate people count but also a better localization of the high-density areas.

In Chapter 5, we extend the temporal consistency in previous chapter to general surveillance video setting. We have shown that implementing a crowd counting algorithm in terms of estimating the people flows and then summing them to obtain people densities is more effective than attempting to directly estimate the densities. This is because it allows us to impose conservation constraints that make the estimates more robust. When optical flow data can be obtained, it also enables us to exploit the correlation between optical flow and people flow to further improve the results. Furthermore, we have demonstrated that spatial and temporal people conservation

can be exploited to train a deep crowd counting model in an active learning fashion, achieving competitive performance with much fewer annotations.

6.2 Limitations and Future Directions

In this section we discuss the main limitations of the proposed methods and suggest potential directions for the future work.

Multi-View Crowd Counting. We use single camera setting for the whole thesis, however in many real-world applications single viewpoint often suffers from heavy occlusion, especially in commercial environment where people are occluded by objects like shelves. Leveraging on multiple cameras would largely ease this situation as people occluded in one viewpoint can be recognized from another viewpoint. Recent work [113] fuses features from multiple viewpoints to estimate the crowd density map in ground plane without leveraging temporal consistency. By enforcing our people flow model in multi-view settings, we are able to reason if the mismatch among different cameras is caused by occlusion or not by checking the people conservation constraint.

Combining Counting with Detection. Compared with detection-based approach, density-map-based crowd counting technique shows better performance for dense crowd. However, if the crowd is sparse and each individual can be clearly detected, the detection-based approach shows even superior performance. Therefore we could combine detection-based approach with our density-map-based one and make them consistency for both sparse and dense crowd. In this way, we are able to robustly localize and count people in various crowd dense levels.

Counting with Better Localization. In this thesis we focus on density-map-based approaches which aims to estimate the crowd density map given input image or video sequence. One drawback of this approach is that the density generally visualize a group of people instead of the detailed location of each instance. This can be solved by leveraging topological constraint [1] or optimal transport [102] to tackle the instance localization in pixel level.

Bibliography

[1] S. Abousamra, M. Hoai, D. Samaras, and C. Chen. Localization in the crowd with topological constraints. In *AAAI Conference on Artificial Intelligence*, 2021.

[2] A. Andriyenko and K. Schindler. Globally Optimal Multi-Target Tracking on a Hexagonal Lattice. In *European Conference on Computer Vision*, pages 466–479, September 2010.

[3] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein Gan. In *arXiv Preprint*, 2017.

[4] C. Arteta., V. Lempitsky, J.A. Noble, and A. Zisserman. Interactive Object Counting. In *European Conference on Computer Vision*, 2014.

[5] C. Arteta., V. Lempitsky, and A. Zisserman. Counting in the Wild. In *European Conference on Computer Vision*, 2016.

[6] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[7] S. Bai, Z. He, Y. Qiao, H. Hu, W. Wu, and J. Yan. Adaptive Dilated Network with Self-Correction Supervision for Counting. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[8] W.H. Beluch, T. Genewein, A. Nürnberger, and J.M. Köhler. The Power of Ensembles for Active Learning in Image Classification. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[9] H. BenShitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking Multiple People Under Global Apperance Constraints. In *International Conference on Computer Vision*, 2011.

[10] H. BenShitrit, J. Berclaz, F. Fleuret, and P. Fua. Multi-Commodity Network Flow for Tracking Multiple People. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1614–1627, 2014.

[11] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua. Multiple Object Tracking Using K-Shortest Paths Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):1806–1819, 2011.

[12] M. V. Borstel, M. Kandemir, P. Schmidt, M. K. Rao, K. Rajamani, J. V. D. Laak, and F. A. Hamprecht. Gaussian Process Density Counting from Weak Supervision. In *European Conference on Computer Vision*, 2016.

[13] G. J. Brostow and R. Cipolla. Unsupervised Bayesian Detection of Independent Motion in Crowds. In *Conference on Computer Vision and Pattern Recognition*, pages 594–601, 2006.

[14] A. Butt and R. Collins. Multiple Target Tracking Using Frame Triplets. In *Asian Conference on Computer Vision*, 2012.

[15] A. Butt and R. Collins. Multi-Target Tracking by Lagrangian Relaxation to Min-Cost Network Flow. In *Conference on Computer Vision and Pattern Recognition*, pages 1846–1853, 2013.

[16] X. Cao, Z. Wang, Y. Zhao, and F. Su. Scale Aggregation Network for Accurate and Efficient Crowd Counting. In *European Conference on Computer Vision*, 2018.

[17] A.B. Chan, Z.S.J. Liang, and N. Vasconcelos. Privacy Preserving Crowd Monitoring: Counting People Without People Models or Tracking. In *Conference on Computer Vision and Pattern Recognition*, 2008.

[18] A.B. Chan and N. Vasconcelos. Bayesian Poisson Regression for Crowd Counting. In *International Conference on Computer Vision*, pages 545–551, 2009.

[19] P. Chattopadhyay, R. Vedantam, R.R. Selvaju, D. Batra, and D. Parikh. Counting Everyday Objects in Everyday Scenes. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[20] K. Chen, C.C. Loy, S. Gong, and T. Xiang. Feature Mining for Localised Crowd Counting. In *British Machine Vision Conference*, page 3, 2012.

[21] Z. Cheng, J. Li, Q. Dai, X. Wu, and A. G. Hauptmann. Learning Spatial Awareness to Improve Crowd Counting. In *International Conference on Computer Vision*, 2019.

[22] O. Chum and J. Matas. Planar Affine Rectification from Change of Scale. In *Asian Conference on Computer Vision*, pages 347–360, 2010.

[23] R.T. Collins. Multitarget Data Association with Higher-Order Motion Models. In *Conference on Computer Vision and Pattern Recognition*, 2012.

[24] C. Dicle, O. I. Camps, and M. Sznaier. The Way They Move: Tracking Multiple Targets with Similar Appearance. In *International Conference on Computer Vision*, 2013.

[25] Y. Fang, B. Zhan, W. Cai, S. Gao, and B. Hu. Locality-Constrained Spatial Transformer Network for Video Crowd Counting. *International Conference on Multimedia and Expo*, 2019.

[26] L. Fiaschi, U. Koethe, R. Nair, and F. Hamprecht. Learning to Count with Regression Forest and Structured Labels. In *International Conference on Pattern Recognition*, pages 2685–2688, 2012.

[27] Y. Gal and Z. Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *International Conference on Machine Learning*, pages 1050–1059, 2016.

[28] A. Gijsberts, M. Atzori, C. Castellini, H. Muller, and B. Caputo. Movement Error Rate for Evaluation of Machine Learning Methods for Semg-Based Hand Movement Classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22:735–744, 2014.

[29] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[30] K. He, X. Zhang, S. Ren, and J. Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In *European Conference on Computer Vision*, 2014.

[31] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[32] Z. He, X. Li, X. You, D. Tao, and Y. Y. Tang. Connected Component Model for Multi-Object Tracking. *IEEE Transactions on Image Processing*, 25(8), 2016.

[33] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[34] Y. Hu, X. Jiang, X. Liu, B. Zhang, and J. Han. Nas-Count: Counting-By-Density with Neural Architecture Search. In *European Conference on Computer Vision*, 2020.

[35] G. Huang, Z. Liu, K.Q. Weinberger, and L. van der Maaten. Densely Connected Convolutional Networks. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[36] H. Idrees, I. Saleemi, C. Seibert, and M. Shah. Multi-Source Multi-Scale Counting in Extremely Dense Crowd Images. In *Conference on Computer Vision and Pattern Recognition*, pages 2547–2554, 2013.

[37] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot, and M. Shah. Composition Loss for Counting, Density Map Estimation and Localization in Dense Crowds. In *European Conference on Computer Vision*, 2018.

[38] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial Transformer Networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[39] X. Jiang, Z. Xiao, B. Zhang, and X. Zhen. Crowd Counting and Density Estimation by Trellis Encoder-Decoder Networks. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[40] X. Jiang, L. Zhang, M. Xu, and T. Zhang. Attention Scaling for Crowd Counting. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[41] D. Kang and A.B. Chan. Crowd Counting by Adaptively Fusing Predictions from an Image Pyramid. In *British Machine Vision Conference*, 2018.

[42] D. Kang, D. Dhar, and A.B. Chan. Incorporating Side Information by Adaptive Convolution. In *Advances in Neural Information Processing Systems*, 2017.

[43] I. Laptev, B. Caputo, C. Schuldt, and T. Lindeberg. Local Velocity-Adapted Motion Events for Spatio-Temporal Recognition. *Computer Vision and Image Understanding*, 108:207–229, 2017.

[44] V. Lempitsky and A. Zisserman. Learning to Count Objects in Images. In *Advances in Neural Information Processing Systems*, 2010.

[45] C.-L. Li, M. Zaheer, Y. Zhang, B. Poczos, and R. Salakhutdinov. Point Cloud GAN. In *arXiv Preprint*, 2018.

[46] Y. Li, X. Zhang, and D. Chen. CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[47] Z. Lin and L.S. Davis. Shape-Based Human Detection and Segmentation via Hierarchical Part-Template Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):604–618, 2010.

[48] C. Liu, X. Weng, and Y. Mu. Recurrent Attentive Zooming for Joint Crowd Counting and Precise Localization. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[49] J. Liu, P. Carr, R. T. Collins, and Y. Liu. Tracking Sports Players with Context-Conditioned Motion Models. In *Conference on Computer Vision and Pattern Recognition*, pages 1830–1837, 2013.

[50] J. Liu, C. Gao, D. Meng, and A.G. Hauptmann1. Decidenet: Counting Varying Density Crowds through Attention Guided Detection and Density Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[51] L. Liu, H. Lu, H. Zou, H. Xiong, Z. Cao, and C. Shen. Weighing Counts: Sequential Crowd Counting by Reinforcement Learning. In *European Conference on Computer Vision*, 2020.

[52] L. Liu, Z. Qiu, G. Li, S. Liu, W. Ouyang, and L. Lin. Crowd Counting with Deep Structured Scale Integration Network. In *International Conference on Computer Vision*, 2019.

[53] M. Liu, W. Buntine, and G. Haffari. Learning How to Actively Learn: A Deep Imitation Learning Approach. In *Annual Meeting of the Association for Computational Linguistics*, 2018.

[54] N. Liu, Y. Long, C. Zou, Q. Niu, L. Pan, and H. Wu. Adcrowdnet: An Attention-Injective Deformable Convolutional Network for Crowd Understanding. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[55] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S.E. Reed, C.-Y. Fu, and A.C. Berg. SSD: Single Shot Multibox Detector. In *European Conference on Computer Vision*, 2016.

[56] W. Liu, K. Lis, M. Salzmann, and P. Fua. Geometric and Physical Constraints for Drone-Based Head Plane Crowd Density Estimation. *International Conference on Intelligent Robots and Systems*, 2019.

[57] W. Liu, M. Salzmann, and P. Fua. Context-Aware Crowd Counting. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[58] W. Liu, M. Salzmann, and P. Fua. Counting People by Estimating People Flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[59] W. Liu, M. Salzmann, and P. Fua. Estimating People Flows to Better Count Them in Crowded Scenes. In *European Conference on Computer Vision*, 2020.

[60] X. Liu, J.V.D.Weijer, and A.D. Bagdanov. Exploiting Unlabeled Data in CNNs by Self-Supervised Learning to Rank. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8), August 2019.

[61] X. Liu, J.V.d. Weijer, and A.D. Bagdanov. Leveraging Unlabeled Data for Crowd Counting by Learning to Rank. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[62] X. Liu, J. Yang, and W. Ding. Adaptive Mixture Regression Network with Local Counting Map for Crowd Counting. In *European Conference on Computer Vision*, 2020.

[63] Y. Liu, L. Liu, P. Wang, P. Zhang, and Y. Lei. Semi-Supervised Crowd Counting via Self-Training on Surrogate Tasks. In *European Conference on Computer Vision*, 2020.

[64] Y. Liu, M. Shi, Q. Zhao, and X. Wang. Point In, Box Out: Beyond Counting Persons in Crowds. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[65] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2015.

[66] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning Transferable Features with Deep Adaptation Networks. In *International Conference on Machine Learning*, pages 97–105, 2015.

[67] C. C. Loy, S. Gong, and T. Xiang. From Semi-Supervised to Transfer Counting of Crowds. In *International Conference on Computer Vision*, 2013.

[68] A. Milan, S. Roth, and K. Schindler. Continuous Energy Minimization for Multitarget Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36:58–72, 2014.

[69] F. Nater, T. Tommasi, H. Grabner, L. V. Gool, and B. Caputo. Transferring Activities: Updating Human Behavior Analysis. In *International Conference on Computer Vision*, 2011.

[70] D. Onoro-Rubio and R.J. López-Sastre. Towards Perspective-Free Object Counting with Deep Learning. In *European Conference on Computer Vision*, pages 615–629, 2016.

[71] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll Never Walk Alone: Modeling Social Behavior for Multi-Target Tracking. In *International Conference on Computer Vision*, 2009.

[72] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects. In *Conference on Computer Vision and Pattern Recognition*, pages 1201–1208, June 2011.

[73] V. Rabaud and S. Belongie. Counting Crowded Moving Objects. In *Conference on Computer Vision and Pattern Recognition*, pages 705–711, 2006.

[74] V. Ranjan, H. Le, and M. Hoai. Iterative Crowd Counting. In *European Conference on Computer Vision*, 2018.

[75] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[76] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems*, 2015.

[77] W. Ren, X. Wang, J. Tian, Y. Tang, and A.B. Chan. Tracking-by-Counting: Using Network Flows on Crowd Density Maps for Tracking Multiple Targets. In *arXiv:2007.09509*, 2020.

[78] M. Rodriguez, I. Laptev, J. Sivic, and J. Audibert. Density-Aware Person Detection and Tracking in Crowds. In *International Conference on Computer Vision*, pages 2423–2430, 2011.

[79] M. Rodriguez, J. Sivic, and I. Laptev. The analysis of high density crowds in videos. *Group and Crowd Behavior for Computer Vision*, 2017.

[80] M. Rodriguez, J. Sivic, I. Laptev, and J. Audibert. Data-driven crowd analysis in videos. In *International Conference on Computer Vision*, 2013.

[81] D.B. Sam, N.N. Sajjan, R.V. Babu, and S. M. Divide and Grow: Capturing Huge Diversity in Crowd Images with Incrementally Growing CNN. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[82] D.B. Sam, N.N. Sajjan, H. Maurya, and R.V. Babu. Almost Unsupervised Learning for Dense Crowd Counting. In *AAAI Conference on Artificial Intelligence*, 2019.

[83] D.B. Sam, S. Surya, and R.V. Babu. Switching Convolutional Neural Network for Crowd Counting. In *Conference on Computer Vision and Pattern Recognition*, page 6, 2017.

[84] G. Schröder, T. Senst, E. Bochinski, and T. Sikora. Optical Flow Dataset and Benchmark for Visual Crowd Analysis. In *International Conference on Advanced Video and Signal Based Surveillance*, 2018.

[85] Z. Shen, Y. Xu, B. Ni, M. Wang, J. Hu, and X. Yang. Crowd Counting via Adversarial Cross-Scale Consistency Pursuit. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[86] M. Shi, Z. Yang, C. Xu, and Q. Chen. Revisiting Perspective Information for Efficient Crowd Counting. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[87] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Advances in Neural Information Processing Systems*, pages 802–810, 2015.

[88] Z. Shi, L. Zhang, Y. Liu, and X. Cao. Crowd Counting with Deep Negative Correlation Learning. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[89] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*, 2015.

[90] V.A. Sindagi and V.M. Patel. Cnn-Based Cascaded Multi-Task Learning of High-Level Prior and Density Estimation for Crowd Counting. In *International Conference on Advanced Video and Signal Based Surveillance*, 2017.

[91] V.A. Sindagi and V.M. Patel. Generating High-Quality Crowd Density Maps Using Contextual Pyramid CNNs. In *International Conference on Computer Vision*, pages 1879–1888, 2017.

[92] V.A. Sindagi and V.M. Patel. Multi-Level Bottom-Top and Top-Bottom Feature Fusion for Crowd Counting. In *International Conference on Computer Vision*, 2019.

[93] V.A. Sindagi, R. Yasarla, D. S. Babu, R. V. Babu, and V.M. Patel. Learning to Count in the Crowd from Limited Labeled Data. In *European Conference on Computer Vision*, 2020.

[94] S. Sinha, S. Ebrahimi, and T. Darrell. Variational Adversarial Active Learning. In *International Conference on Computer Vision*, 2019.

[95] D. Sun, X. Yang, M. Liu, and J. Kautz. Pwc-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[96] J.W. Suurballe. Disjoint Paths in a Network. *Networks*, 1974.

[97] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. In *Conference on Computer Vision and Pattern Recognition*, pages 1–9, June 2015.

[98] C. Vogel, S. Roth, and K. Schindler. 3D Scene Flow Estimation with a Rigid Motion Prior. In *International Conference on Computer Vision*, 2011.

[99] E. Walach and L. Wolf. Learning to Count with CNN Boosting. In *European Conference on Computer Vision*, 2016.

[100] J. Wan and A. B. Chan. Adaptive Density Map Generation for Crowd Counting. In *International Conference on Computer Vision*, 2019.

[101] J. Wan, W. Luo, B. Wu, A. B. Chan, and W. Liu. Residual Regression with Semantic Prior for Crowd Counting. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[102] B. Wang, H. Liu, D. Samaras, and M. Hoai. Distribution Matching for Crowd Counting. In *Advances in Neural Information Processing Systems*, 2020.

[103] Q. Wang, J. Gao, W. Lin, and Y. Yuan. Pixel-Wise Crowd Understanding via Synthetic Data. *International Journal of Computer Vision*, 2020.

[104] X. Wang, B. Wang, and L. Zhang. Airport Detection in Remote Sensing Images Based on Visual Attention. In *International Conference on Neural Information Processing*, 2011.

[105] B. Wu and R. Nevatia. Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors. In *International Conference on Computer Vision*, 2005.

[106] F. Xiong, X. Shi, and D. Yeung. Spatiotemporal Modeling for Crowd Counting in Videos. In *International Conference on Computer Vision*, pages 5161–5169, 2017.

[107] H. Xiong, H. Lu, C. Liu, L. Liu, Z. Cao, and C. Shen. From Open Set to Closed Set: Counting Objects by Spatial Divide-And-Conquer. In *International Conference on Computer Vision*, 2019.

[108] C. Xu, K. Qiu, J. Fu, S. Bai, Y. Xu, and X. Bai. Learn to Scale: Generating Multipolar Normalized Density Maps for Crowd Counting. In *International Conference on Computer Vision*, 2019.

[109] Z. Yan, Y. Yuan, W. Zuo, X. Tan, Y. Wang, S. Wen, and E. Ding. Perspective-Guided Convolution Networks for Crowd Counting. In *International Conference on Computer Vision*, 2019.

[110] Y. Yang, G. Li, Z. Wu, L. Su, Q. Huang, and N. Sebe. Reverse Perspective Network for Perspective-Aware Object Counting. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[111] C. Zhang, H. Li, X. Wang, and X. Yang. Cross-Scene Crowd Counting via Deep Convolutional Neural Networks. In *Conference on Computer Vision and Pattern Recognition*, pages 833–841, 2015.

[112] L. Zhang, Y. Li, and R. Nevatia. Global Data Association for Multi-Object Tracking Using Network Flows. In *Conference on Computer Vision and Pattern Recognition*, 2008.

[113] Q. Zhang and A. B. Chan. Wide-Area Crowd Counting via Ground-Plane Density Maps and Multi-View Fusion CNNs. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[114] S. Zhang, G. Wu, J.P. Costeira, and J.M.F. Moura. FCN-rLSTM: Deep Spatio-Temporal Neural Networks for Vehicle Counting in City Cameras. In *International Conference on Computer Vision*, 2017.

[115] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma. Single-Image Crowd Counting via Multi-Column Convolutional Neural Network. In *Conference on Computer Vision and Pattern Recognition*, pages 589–597, 2016.

[116] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid Scene Parsing Network. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[117] Z. Zhao, M. Shi, X. Zhao, and L. Li. Active Crowd Counting with Limited Supervision. In *European Conference on Computer Vision*, 2020.

Weizhe Liu
weizhe.liu@epfl.ch | +41 21 693 23 07 | https://weizheliu.github.io

RESEARCH INTERESTS

Crowd Analysis (Counting, Localization and Motion), Video Understanding, Action Recognition, Semantic Segmentation, Domain Adaptation, Learning with Less Supervision

EDUCATION

École Polytechnique Fédérale de Lausanne (EPFL) — Lausanne, Switzerland
Ph.D. in Computer Science — June 2017 – Oct. 2021
Advisor: Prof. Pascal Fua
Research Group: Computer Vision Laboratory

University of California, Los Angeles (UCLA) — Los Angeles, US
Visiting Scholar — Sept. 2016 – Mar. 2017
Advisor: Prof. Stefano Soatto
Research Group: UCLA Vision Lab

École Polytechnique Fédérale de Lausanne (EPFL) — Lausanne, Switzerland
M.Sc. in Communication Systems — Sept. 2014 – Apr. 2017
Title of Thesis: Active Perception Using Recurrent Neural Networks
Thesis Advisor: Prof. Stefano Soatto and Prof. Pascal Fua

University of Electronic Science and Technology of China (UESTC) — Chengdu, China
B.Eng in Electronic and Information Engineering — Sept. 2010 – July 2014
Title of Thesis: Video Compressing With H.264
Thesis Advisor: Prof. Feng Fan

WORK EXPERIENCE

Microsoft — Zurich, Switzerland
Research Intern — Apr. 2021 – June 2021
Project: Self-Supervised Video Alignment for Action Recognition
Mentor: Dr. Bugra Tekin

Amazon — Graz, Austria
Research Intern — July 2020 – Oct. 2020
Project: Semi-Supervised Domain Adaptation for Semantic Segmentation
Mentor: Dr. Christian Leistner

NVISO — Lausanne, Switzerland
Computer Vision Engineer Intern — Feb. 2016 – Aug. 2016
Project: Lightweight Caffe Framework for Mobile Devices
Mentor: Timothy llewellynn and Dr. Matteo Sorci

TEACHING

- CS-233(a), Introduction to machine learning(BA3)
- CS-233(b), Introduction to machine learning (BA4)
- MATH-233, Probabilities and statistics
- MATH-101(e), Analysis I

PROFESSIONAL SERVICES

Reviewer of major computer vision conferences (*CVPR*, *ICCV*, *ECCV*) and journals (*PAMI*, *IJCV*, *TIP*).

PREPRINTS

[1] **W. Liu**, D. Ferstl, S. Schulter, L. Zebedin, P. Fua and C. Leistner. Domain Adaptation for Semantic Segmentation via Patch-Wise Contrastive Learning. arXiv:2104.11056.

[2] **W. Liu**, N. Durasov and P. Fua. Leveraging Self-Supervision for Cross-Domain Crowd Counting. arXiv:2103.16291.

PUBLISHED

[1] **W. Liu**, M. Salzmann and P. Fua. Counting People by Estimating People Flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (PAMI), 2021.

[2] **W. Liu**, M. Salzmann and P. Fua. Estimating People Flows to Better Count Them in Crowded Scenes. *The European Conference on Computer Vision* (ECCV), 2020.

[3] **W. Liu**, K. Lis, M. Salzmann and P. Fua. Geometric and Physical Constraints for Drone-Based Head Plane Crowd Density Estimation. *The IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS), 2019.

[4] **W. Liu**, M. Salzmann and P. Fua. Context-Aware Crowd Counting. *The IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2019.

RELEVANT SKILLS
Programming Language: Python, MATLAB, C++
Software Framework: PyTorch, OpenCV, TensorFlow, Caffe