Measuring the effect of nuisance variables on classifiers

Alhussein Fawzi
alhussein.fawzi@epfl.ch
Pascal Frossard
pascal.frossard@epfl.ch

Signal Processing Laboratory (LTS4)
Ecole Polytechnique Fédérale de
Lausanne (EPFL)
Lausanne, Switzerland

Abstract

In real-world classification problems, *nuisance variables* can cause wild variability in the data. Nuisance corresponds for example to geometric distortions of the image, occlusions, illumination changes or any other deformations that do not alter the ground truth label of the image. It is therefore crucial that designed classifiers are robust to nuisance variables, especially when these are deployed in real and possibly hostile environments. We propose in this paper a probabilistic framework for efficiently *estimating* the robustness of state-of-the-art classifiers and *sampling* problematic samples from the nuisance space. This allows us to visualize and understand the regions of the nuisance space that cause misclassification, in the perspective of improving robustness. Our probabilistic framework is applicable to arbitrary classifiers and potentially high-dimensional and complex nuisance spaces. We illustrate the proposed approach on several classification problems and compare classifiers in terms of their robustness to nuisances. Moreover, using our sampling technique, we visualize problematic regions in the nuisance space and infer insights into the weaknesses of classifiers as well as the features used in classification (e.g., in face recognition). We believe the proposed analysis tools represent an important step towards understanding large modern classification architectures and building architectures with better robustness to nuisance.

1 Introduction

Image classification has recently witnessed key advances in performance on many challenging benchmarks [10, 16, 17, 20]. Despite these advances, classification systems are however often regarded as black box models that lead only to limited understanding of the weaknesses of a given model. We focus here on a key property of classifiers, that is their ability to factor out *nuisance variables*. Nuisance accounts for variability that has no effect on the result of the task, and should be ideally factored out of the classification system. For example, nuisance may correspond to changes in illumination, occlusion or standard local geometric transformation of the image. It is well known that humans are excellent at removing irrelevant information in a visual task, such as recognizing an object independently of its viewpoint or with occluding objects, while it is unclear to which extent state-of-the-art classifiers can factor out such complex nuisances. To have a better understanding of how modern classifiers deal with nuisance variables, it becomes crucial to develop *generic methods* to measure and visualize the effect of nuisance on classifiers.

We propose in this paper a general probabilistic framework for assessing and analyzing the robustness of classifiers to nuisance factors. The outcomes of the proposed framework are two-fold: the *estimation* of the robustness of classifiers to arbitrary nuisances and the *sampling* from problematic regions in the nuisance space for visualizing and possibly improving the robustness to nuisances. Specifically, we first propose a formal definition of the *average robustness to nuisance*, and provide a provably efficient Monte-Carlo estimate. In a second step, we focus on *problematic* regions of the nuisance space, where the classifier outputs low confidence scores for highly probable nuisance values, and propose a MCMC sampling mechanism to quickly reach such regions of the nuisance space. This allows us to visualize problematic samples for a given classifier, and gain further insights into regions of the nuisance space which cause misclassification. Such a sampling mechanism can also potentially be very valuable to improve the robustness of classifiers. Our framework is generic and can be applied to any parametrizable nuisance space and classifier. To illustrate the proposed framework, we apply it to several classification architectures, three classification datasets and three nuisance spaces. We quantify in particular the effect of data augmentation, dropout, spatial transformer network layers [8] on the robustness of CNNs, and compare state-of-the-art deep neural networks trained on natural image datasets in terms of their robustness to standard nuisances. Our results provide insights into the important features used by the classifier to distinguish between classes, through the visualization of the nuisances that transform an image to a different class. Our experiments also demonstrate that state-of-the-art classifiers are only mildly robust to standard nuisances, and that more effort should therefore be spent to improve this robustness.

Following the major successes of deep visual representations, several empirical [1, 4, 9, 11] studies, theoretical analysis [18] and visualization tools [3, 12] have been proposed to achieve a better understanding of the *geometric* properties and viewpoint invariance of deep visual representations. Unfortunately, these analysis works are either restricted to specific nuisance spaces (e.g., low-dimensional geometric transformations or mathematical groups), particular classification methods (deep convolutional neural networks), or do not offer mechanisms to visualize and sample from problematic nuisance regions. Other forms of robustness to perturbations have recently been analyzed in [5, 15, 19], where *minimal* additive perturbations that are sufficient to change the label of the classifier are sought. This worst-case robustness definition has later been applied to geometric transformations to assess the invariance of classifiers to rigid transformations [4]. The sampling paradigm we develop in this paper significantly departs from these approaches, where we *explore* the space of nuisance variables that cause misclassification, rather than focusing only on the minimal perturbation or geometric transformation. Finally, it should be noted that new classification architectures have been proposed with the goal of improving the robustness to various deformations in the data [7, 8, 13]. In that context, we believe that our framework can readily be used in order to *quantitatively* evaluate these classifiers in terms of their robustness, and more importantly, contribute to improving the robustness of classifiers to nuisances by leveraging the proposed sampling mechanism. The code of the algorithms will be made available online on the project webpage.[1]

[1] https://sites.google.com/site/classifiernuisance/

2 Measuring the effect of nuisance variables

2.1 Definitions

We consider an arbitrary classifier that is provided through its conditional distribution $p_{\text{cl}}(c|x)$, which represents the probability that an image x is classified as c by the classifier.[2] In neural network architectures, this discrete conditional distribution $p_{\text{cl}}(\cdot|x)$ corresponds to the probability vector that can be read at the last layer of the neural network (i.e., after the softmax layer), after a feedforward pass of the input x.

Let $\mathcal{T}$ be a set of nuisances. The set $\mathcal{T}$ can for example represent the set of affine transformations, diffeomorphisms, or occlusions that might corrupt the data. For a particular element in the nuisance set $\theta \in \mathcal{T}$, we define $T_\theta x$ to be the image x transformed by θ. We adopt a Bayesian framework and equip the nuisance space $\mathcal{T}$ with a *prior* probability distribution $p_{\mathcal{T}}(\theta)$ that captures our region of interest in the nuisance space. For example, when $\mathcal{T}$ denotes the occlusion nuisance set, $p_{\mathcal{T}}(\theta)$ might take large values for small occlusions (covering small parts of the image), and smaller values for large occlusions. In some cases, the prior distribution $p_{\mathcal{T}}(\theta)$ might depend on the image; hence, for the sake of generality, we denote our prior distribution by $p_{\mathcal{T}}(\theta|x)$.

We now define a quantity that allows us to measure the robustness of a classifier with respect to a nuisance set $\mathcal{T}$. Consider an image x with ground truth label $\ell(x)$. The quantity $p_{\text{cl}}(\ell(x)|x)$ reflects the confidence that x is classified as $\ell(x)$, and therefore should be large when the classifier is accurate. For a given nuisance $\theta \in \mathcal{T}$, the expression $p_{\text{cl}}(\ell(x)|\theta,x) := p_{\text{cl}}(\ell(x)|T_\theta x)$ gives the probability that the transformed image $T_\theta x$ is also classified as the ground truth $\ell(x)$. For a classifier to be robust, this quantity should be large for typical θ. We define the robustness $\mu_{\mathcal{T}}(x)$ as the average of this quantity, weighted by $p_{\mathcal{T}}(\theta|x)$:

$$\mu_{\mathcal{T}}(x) := \int_\theta p_{\text{cl}}(\ell(x)|\theta,x)p_{\mathcal{T}}(\theta|x)d\theta = \mathop{\mathbb{E}}_{\theta \sim p_{\mathcal{T}}(\cdot|x)}(p_{\text{cl}}(\ell(x)|\theta,x)).$$

Note that our quantity $\mu_{\mathcal{T}}(x)$ strongly depends on the prior distribution $p_{\mathcal{T}}(\theta|x)$; a classifier with a large $\mu_{\mathcal{T}}(x)$ will have high classification confidence in highly probable regions of the nuisance space, but $\mu_{\mathcal{T}}(x)$ is only mildly affected by the classifier confidence in low probability regions of $\mathcal{T}$. In a Bayesian inference setting, $\mu_{\mathcal{T}}(x)$ is called the *marginalized* likelihood, where the likelihood term $p_{\text{cl}}(\ell(x)|\theta,x)$ is marginalized over $p_{\mathcal{T}}$.

Given a data distribution p_d, we define the *global* robustness to nuisance variables in $\mathcal{T}$ as the average of $\mu_{\mathcal{T}}(x)$, i.e.,

$$\rho_{\mathcal{T}} := \int_x \mu_{\mathcal{T}}(x)p_{\text{d}}(x)dx = \mathop{\mathbb{E}}_{x\sim p_{\text{d}}}\left(\mathop{\mathbb{E}}_{\theta\sim p_{\mathcal{T}}(\cdot|x)}[p_{\text{cl}}(\ell(x)|\theta,x)]\right)$$

It should be noted that the quantities $\mu_{\mathcal{T}}$ and $\rho_{\mathcal{T}}$ are bounded between 0 and 1. The global robustness $\rho_{\mathcal{T}}$ measures the average confidence that typical images perturbed with nuisances chosen according to the prior distribution $p_{\mathcal{T}}(\theta|x)$ are classified as $\ell(x)$.

[2] Unlike many works that assume that the mapping from the input x to the label is deterministic, it is assumed here to be a probabilistic mapping defined by the conditional distribution $p_{\text{cl}}(c|x)$. The probabilistic definition is more general as the deterministic case corresponds to $p_{\text{cl}}(c|x) = 1$ for the correct label, and 0 otherwise.

2.2 Estimation of the global robustness score

The global robustness measure $\rho_{\mathcal{T}}$ is a continuous quantity that involves an integration over the image space, as well as the nuisance space. We estimate these quantities using a Monte Carlo approximation method, and define the empirical quantities $\hat{\mu}_{\mathcal{T}}$ and $\hat{\rho}_{\mathcal{T}}$ as

$$\hat{\mu}_{\mathcal{T}}(x) = \frac{1}{N} \sum_{i=1}^{N} p_{\text{cl}}(\ell(x)|\theta_i, x), \text{ with } \theta_i \overset{\text{iid}}{\sim} p_{\mathcal{T}}(\theta|x), \tag{1}$$

$$\hat{\rho}_{\mathcal{T}} = \frac{1}{M} \sum_{j=1}^{M} \hat{\mu}_{\mathcal{T}}(x_j), \text{ with } x_j \overset{\text{iid}}{\sim} p_{\text{d}}. \tag{2}$$

$\mu_{\mathcal{T}}(x)$ is approximated by the average of the likelihood $p_{\text{cl}}(\ell(x)|\theta_i, x)$ over iid samples generated from the prior distribution $p_{\mathcal{T}}(\theta|x)$. The global robustness measure is then naturally defined as the empirical average of $\hat{\mu}_{\mathcal{T}}(x_j)$, over iid samples from the data distribution.

The computation of Eq. (1, 2) involves the transformation and classification of NM samples. For computational purposes, it is therefore crucial that the empirical quantities approximate the true quantities while keeping a small number of samples. The following result derives theoretical guarantees on the approximation error with respect to the number of samples N and M.

Theorem 1. *Let $t > 0$, and $\delta \in (0,1)$. We have $|\hat{\rho}_{\mathcal{T}} - \rho_{\mathcal{T}}| \leq t$ with probability exceeding $1 - \delta$ as long as $M \geq \frac{\ln(2/\delta)}{2t^2}$. Moreover, when the prior distributions are data-independent (i.e., $p_{\mathcal{T}}(\theta|x) = p_{\mathcal{T}}(\theta)$), the above condition becomes $NM \geq \frac{\ln(2/\delta)}{2t^2}$.*

The proof of the theorem, which follows from the concentration of measure of bounded random variables, is deferred to the supplementary material due to space constraints.

For prior distributions on nuisance spaces that are independent of the datapoint x, the above result shows that, by choosing N and M in the order of 100, one can obtain very accurate estimates for $\hat{\rho}_{\mathcal{T}}$. When the nuisance prior is data-dependent, the worst-case result becomes independent of N, and one needs more samples to derive accurate estimates. In many cases of interest however, the independent case practically applies as the prior distribution does not significantly differ for different images. It should finally be noted that the bounds in Theorem 1 do not depend on the dimension of the nuisance space; this shows that the approximate quantity $\hat{\rho}_{\mathcal{T}}$ can be very accurate (for moderately large N and M) even for high dimensional nuisance spaces.

2.3 Estimation of the problematic nuisances

While $\rho_{\mathcal{T}}$ measures the *average* likelihood of the classifier (i.e., confidence of correct classification, when nuisance samples are drawn from the prior distribution), it is also crucial to visualize and understand the *problematic* regions of the nuisance space where the classifier has low confidence on transformed images. The problematic regions of the nuisance space are mathematically described by the *posterior* distribution $p_{\text{cl}}(\theta|\overline{\ell(x)}, x)$, where we define $p_{\text{cl}}(\overline{\ell(x)}|\theta, x) = 1 - p_{\text{cl}}(\ell(x)|\theta, x)$ to be the probability that $T_\theta x$ is *not* classified as $\ell(x)$. Sampling from this posterior distribution allows us to "diagnose" the set of nuisance parameters that can cause classification errors. Using the Bayes rule, the posterior distribution can be written as the normalized product of the likelihood and prior distribution $p_{\text{cl}}(\theta|\overline{\ell(x)}, x) = \frac{1}{Z} p_{\text{cl}}(\overline{\ell(x)}|\theta, x) p_{\mathcal{T}}(\theta|x)$, with Z the normalizing constant. It should be noted

Algorithm 1 Metropolis algorithm for sampling from $p_{\text{cl}}(\theta|\overline{\ell(x)},x)$

```
**Initialization:** Start with a randomly initialized sample in the nuisance set $\theta^{(0)} \in \mathcal{T}$.
**For each iteration $s$ of the random walk on the nuisance space $\mathcal{T}$, do:**
    Draw a sample $\theta' \sim q(\theta|\theta^{(s)})$.
    Let $p_{\text{accept}} = \min\left(1, \frac{p_{\text{cl}}(\overline{\ell(x)}|\theta',x)p_{\mathcal{T}}(\theta'|x)q(\theta^{(s)}|\theta')}{p_{\text{cl}}(\overline{\ell(x)}|\theta^{(s)},x)p_{\mathcal{T}}(\theta^{(s)}|x)q(\theta'|\theta^{(s)})}\right)$.
    Generate a random uniform sample in $u \in [0,1]$.
    If $u \leq p_{\text{accept}}$, $\theta^{(s+1)} \leftarrow \theta'$; otherwise, $\theta^{(s+1)} \leftarrow \theta^{(s)}$.
```

that this posterior distribution is typically a complex high dimensional distribution, where specialized sampling algorithms do not apply.

To sample from this posterior distribution, we adopt here the celebrated Metropolis MCMC method for sampling from high dimensional distributions [14]. The sample values are produced iteratively, where the distribution of the next sample depends only on the current sample value (hence making the samples sequence a Markov Chain). At each iteration, the algorithm picks a candidate for the next sample by sampling from a *proposal distribution* q, which guides the exploration of the nuisance space $\mathcal{T}$. Then, with some probability p_{accept}, the candidate is either accepted, in which case the candidate value is used in the next iteration, or rejected. The acceptance probability is controlled by the ratio between the probability of the posterior distribution at the candidate sample to that of the current sample. The algorithm is summarized in Algorithm 1.

In practice, we set the proposal distribution $q(\cdot|\theta) \sim \mathcal{N}(\theta, \sigma_{\text{prop}}^2 I)$. It should be noted that the above algorithm can be applied to any parametrizable nuisance space $\mathcal{T}$, and any prior distribution $p_{\mathcal{T}}$ (potentially *complex* prior distributions where sampling is difficult) in order to find problematic samples of the nuisance space. Fig. 1 illustrates the samples drawn using the Metropolis algorithm when $\mathcal{T}$ is the set of 2D translations, and the nuisance space is equipped with a Gaussian prior centered at 0. An arbitrary digit image together with a baseline classifier was used for the sake of this illustrative example. It can be seen that samples obtained with Metropolis confine to highly probable regions of the nuisance space (these correspond to nuisance parameters with *low* classification confidence). In particular, it should be noted that the Metropolis method relying on a Markov Chain random walk for sampling is much more efficient than the standard approach where independent samples are drawn from $p_{\mathcal{T}}(\theta|x)$, and accepted or rejected depending on the values of their likelihood. This Metropolis method is therefore particularly suited to our framework, as it can efficiently sample "problematic samples", even if $\mu_{\mathcal{T}}(x) \approx 1$.

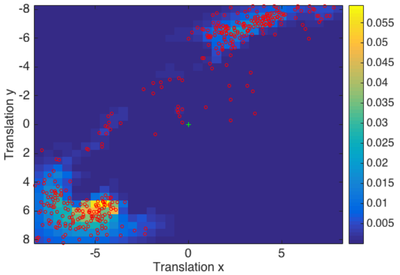

Figure 1: Example map of the (unnormalized) posterior distribution $p_{\text{cl}}(\theta|\overline{\ell(x)},x)$ when $\mathcal{T}$ = 2d translations. We overlay samples obtained using the Metropolis MCMC method.

Model	Test error (%)	$\hat{\rho}_{\mathcal{T}}$ ($\alpha = 100$)	$\hat{\rho}_{\mathcal{T}}$ ($\alpha = 50$)	$\hat{\rho}_{\mathcal{T}}$ ($\alpha = 10$)
CNN-1	1.26	0.90	0.76	0.30
+ **Dropout**	0.88	0.90	0.77	0.31
+ **DA**	1.04	0.93	0.85	0.44
+ **STN**	0.93	**0.96**	**0.90**	0.52
CNN-2	1.16	0.94	0.83	0.36
+ **Dropout**	**0.68**	0.93	0.82	0.37
+ **DA**	1.09	0.94	0.87	0.48
+ **STN**	0.79	**0.96**	**0.90**	**0.53**

Table 1: Robustness to affine transformations of several networks on the MNIST dataset. Each network is trained for 50 epochs.

3 Experimental evaluation

3.1 MNIST handwritten digits

We evaluate in this section different classifiers in terms of their robustness to the set $\mathcal{T}$ of affine transformations. We parametrize the elements in $\mathcal{T}$ with vectors $\theta \in \mathbb{R}^6$ representing the column-reshaped 2×3 standard matrix representations of affine transformations. We consider a Gaussian prior distribution on $\mathcal{T}$ given by $p_{\mathcal{T}} = \mathcal{N}(\mathbf{1}, \Sigma)$, where $\mathbf{1}$ is the identity affine transformation, and $\Sigma \in \mathbb{R}^{6 \times 6}$ is a covariance matrix that penalizes large changes in the appearance of the image. Using differential geometric considerations that we defer to the supplementary material due to space constraints, we set $\Sigma = (\alpha G)^{-1}$, where α is a parameter that controls the magnitude of the transformation (lower α imply larger transformations), and G denotes a Riemannian metric at $\mathbf{1}$ of the manifold $\mathcal{T}$. We also refer to the supplementary material for visualizations of the transformed versions of arbitrary MNIST images, with transformations drawn from the prior distribution using different values of α.

We consider two baseline CNN architectures on the MNIST task, CNN-1 and CNN-2, of respectively 1 and 2 hidden layers. We then consider the following modifications of these baseline neural networks:

- **Dropout regularization**: We use a dropout regularization (with probability $p = 0.5$) at the last fully connected layer of the network,
- **Data Augmentation (DA)**: At the training stage, we apply a small random translation to the samples with probability 0.1. In other words, we randomly translate 10% of the samples at the training stage.
- **Spatial Transformer Network (STN)** [8]: We use a model where the localization network is a two layer CNN which operates on the image input. The output from the localization network is a 6 dimensional vector specifying the parameters of the affine transformation. This network is trained with data augmentation.

Dropout, DA and STN are often used in order to improve the classification performance. The goal here is to see the effect of these techniques on the robustness to nuisance factors.

Table 1 reports the affine robustness $\hat{\rho}_{\mathcal{T}}$ with $N = M = 1000$ for the different networks for three transformation regimes (mild, medium and severe transformations respectively obtained by setting $\alpha = 100, 50, 10$). By comparing CNN-1 and CNN-2, it can be seen that increasing the number of layers leads to a better affine invariance of the model. This result is in line with the conclusions of [4] showing that an increase in the number of layers

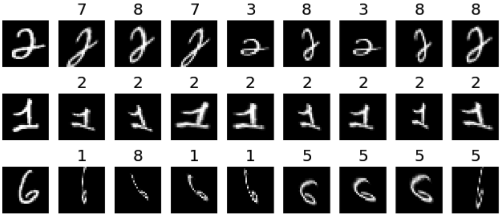

Figure 2: Samples drawn from the posterior distribution $p(\theta|\overline{\ell(x)},x)$ with $\alpha = 100$. On the left, the original image, and then the transformed images with nuisances sampled from the posterior distribution for the CNN-2 with Spatial Transformer Network. The estimated label by the classifier of each transformed image is shown on top of each image. All shown images are misclassified by the classifier.

of a deep convolutional network leads to improved robustness to similarity transformations. While dropout regularization leads to significant improvement in test accuracy, it has barely any effect on the robustness of the classifier to affine transformations. This shows that robustness and test accuracy capture two different properties of the classifier. In fact, while the robustness property measures the effect of nuisance variables that might occur in real-world applications on the classification function, the test set usually contains a restricted set of images following the same distribution as the training set. Conversely, data augmentation (with translated samples) has led in this example to a decrease in the test accuracy, while boosting the robustness to transformations. Moreover, the addition of STN layers also improves the robustness of classifiers to transformations in the data.

Among the tested classifiers, CNN-2-STN has the maximum robustness for all parameters α, with an robustness score larger than 0.9 for mild and medium transformations ($\alpha = 100, 50$). In other words, the classifier correctly classifies transformed samples with confidence surpassing 90%. Nevertheless, despite these large *average* scores, this same network can wrongly classify images that are however quite easily identifiable by a human observer. To see this, we show in Fig. 2 transformed images with samples drawn from the posterior distribution $p_{\text{cl}}(\theta|\overline{\ell(x)},x)$ using the Metropolis algorithm.[3] Quite interestingly, these samples have a large variation, thereby showing multiple "flaws" of the classifier. For example, relatively small transformations of a digit 2 can cause it to be a 7, 8 or 3. This shows the existence of many "directions" that potentially cause the classifier to misclassify.

3.2 Natural images classification and face recognition

We now conduct experiments on deep neural networks that are trained on the ImageNet challenge dataset. Specifically, we consider 4 different pre-trained networks: VGG-CNN-S [2], VGG-16, VGG-19 [17], and GoogLeNet [20]. We evaluate the robustness of these networks to *piecewise* affine transformations. Specifically, the image is divided into cells, and each cell undergoes a different affine transformation. We parametrize the transformations using motion vectors defined for regularly spaced control points in the image. More precisely, a transformation is parametrized by a set of motion vectors stacked in an array $\mathbf{V} \in \mathbb{R}^{2\times L}$, where L defines the number of control points. We then define a prior distribution

[3] We post-processed the samples obtained using Metropolis (section 2.3) by keeping only the samples having a label different than $\ell(x)$. The depicted samples are randomly chosen from this set.

$p_{\mathcal{T}} = \mathcal{N}(\mathbf{0}_{2L}, \Sigma)$, with a covariance matrix Σ whose correlations decay with the distance between control points, and $\mathbf{0}_{2L}$ denotes the zero motion vector. This distribution, which forces nearby control points to have similar motion vectors, incorporates a smoothness constraint on the set of transformations, and results in having well-behaved and natural transformations. It should be noted that [6] defined a similar prior distribution on motion fields in a different context. We refer to the supplemental material for the illustration of images transformed with nuisance parameters sampled from the introduced prior distribution.

We report the robustness measures $\hat{\rho}_{\mathcal{T}}$ of the different networks in Table 2, for $M = 200$ and $N = 100$. It can be noted that VGG-CNN-S is slightly worse than other networks in terms of robustness to piecewise affine transformations. This confirms once again the result highlighted in the previous section, namely that depth improves the robustness to nuisance factors (in particular piecewise affine transformations), as VGG-16, 19 and GoogLeNet contain substantially more layers than VGG-CNN-S. The overall scores shown in Table 2 show however that these state-of-the-art networks correctly classify samples with confidence lower than 70%, for sufficiently small piecewise affine transformations of the data.

VGG-CNN-S	VGG-16	VGG-19	GoogLeNet
0.62	0.68	0.68	0.67

Table 2: Robustness to piecewise affine transf. of different networks trained on ImageNet

We visualize images with nuisances sampled from the posterior $p_{\text{cl}}(\theta|\overline{\ell(x)}, x)$ in Fig. 3 for the different networks. For some examples, a "natural" transformation of the image leads to a label change: observe that the "Gyromitra" is indeed transformed to be visually similar to an image representing a "hen". These examples provide insights into *the concepts* that the deep network uses to discriminate between the classes. In particular, observe that the required nuisance parameter θ to transform a "white wolf" onto an "arctic fox" or "Samoyed" is rather intuitive for a human. In particular, note that the deep network heavily relies on the deformation of the "nose" cue in order to change the estimated label; this shows that the deep network uses this cue in order to distinguish between these neighbouring classes. It should be noted however that for some images, relatively small transformations are sufficient to change the estimated class to labels that are very different from a human perspective (e.g., lampshade → sea slug, necklace, ...). This shows deficiencies in the concepts learned by these classifiers, and that the context of the image is probably not sufficiently used to infer the label (e.g., the context of a scene representing a lamp shade is very different from sea slug). In fact, while it is possible to modify the geometric aspect of an object (e.g., lampshade) using a nuisance transformation from $\mathcal{T}$, the overall scene context (characterized by the shadings, neighbouring objects, etc...) is much more difficult to alter and should ideally be detected by the classifier to achieve robustness.

We finally consider a face recognition application, where we consider the very recent VGG-Face classifier from [16], and measure the robustness of this classifier to simple *occlusions*. Specifically, we consider a nuisance set $\mathcal{T}$ where b occlusion rectangles corrupt the images: any pixel belonging to one of the rectangles is "erased" and set to zero. We consider a prior probability distribution on this nuisance space that penalizes the total area of occluded pixels. Specifically, we set $p_{\mathcal{T}}(\theta) \propto \exp\left(-O_p/\sigma^2\right)$, where O_p is the number of occluded pixels by the b rectangles, and $\theta \in \mathbb{R}^{4b}$ is a parametrization of the state where each rectangle is parametrized with 4 scalars (upper left and lower right point). In the experiments, we set $\sigma = 2000$, $b = 3$. For the Metropolis algorithm, we use a Gaussian proposal with standard

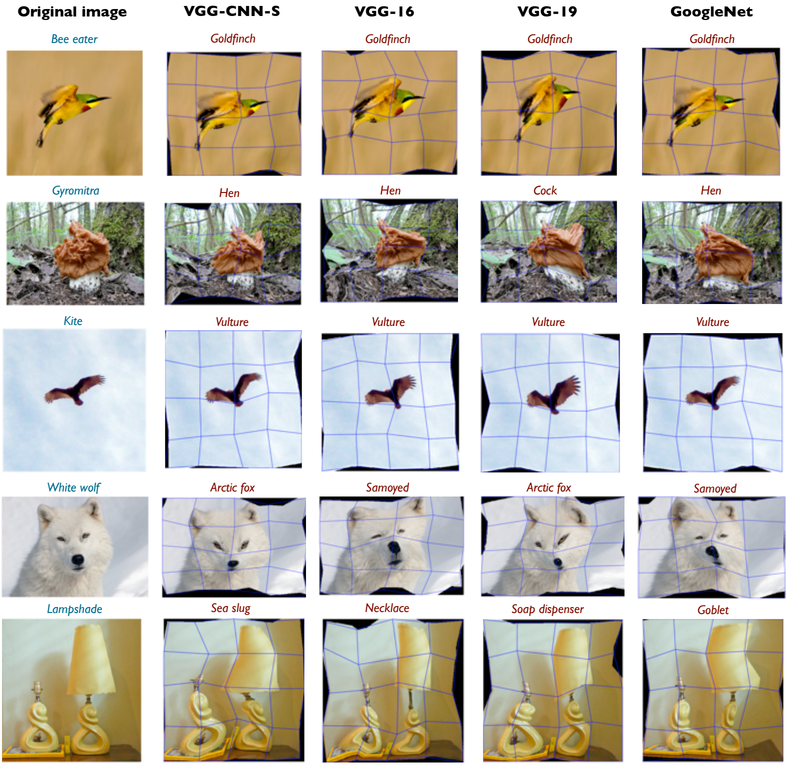

Figure 3: Robustness of networks trained on ImageNet challenge to piecewise affine transformations. The left column represents the original image, and the remaining columns show transformed images sampled from the posterior distribution. On top of each image, we display the label estimated by the classifier. The post-processing step of footnote 3 was applied.

deviation $\sigma_{\text{prop}} = 5$, and set the number of iterations to 1000. The samples are shown in Fig. 4. Interestingly, it can be seen that with relatively small occluding boxes, one can change the estimated label of the classifier. More surprising, these simple occlusions can cause *trivial* errors in the estimated label (e.g., *Aamir Khan* → *Craig Robinson*, or *Daniel Craig* → *Anna Gunn*). This lack of robustness is specifically problematic in a face recognition system as it can be exploited by intruders for fraudulous identification in systems using face recognition. The proposed sampling tool is important to assess the robustness to such nuisances, and reveal the weaknesses of such classifiers before their deployment in possibly hostile environments. Moreover, similarly to the visualization in Fig. 3, one can infer insights on the features used in the face recognition. Specifically, we observed that in many cases, the classifier changes label by adding a relatively small occluding box on the person's nose, which suggests that this represents an important feature in this automatic face recognition system.

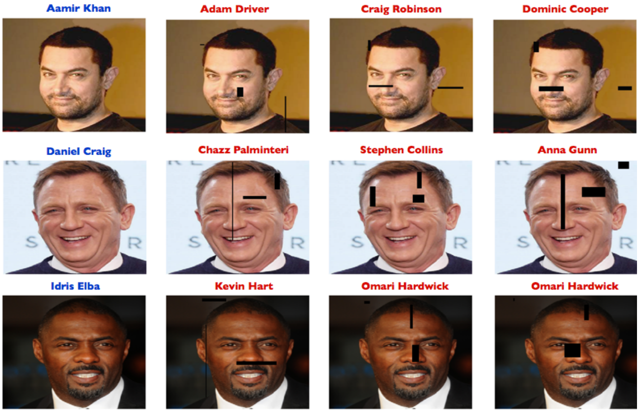

Figure 4: Robustness of VGG-Faces classifier to artificial occlusion. Left column: original image, with correct label. Columns 2 to 4 are samples from the posterior distribution. On top of each image, we indicate the *estimated* label. The post-processing step in footnote 3 was applied to choose the samples for visualization.

4 Discussion

We proposed in this paper a simple and generic probabilistic framework for measuring the average robustness to nuisance variables, as well as for sampling problematic nuisance variables. Our framework can deal with *any* type of parametrizable nuisance factors, as long as a prior distribution that defines the region of interest on this space is defined. The proposed tool permits to generate samples that represent "weak points" of the classifier. We expect that this framework will be used in applications where robustness to (possibly hostile) nuisance perturbations is essential (e.g., security applications). The proposed approach can also be used to derive insights on the features used for classification. Finally, we believe the current work provides opportunity to further improve the robustness of classifiers, as the samples revealing the "weak points" of the classifier can potentially be used to re-train the classifier.

References

[1] A. Bakry, M. Elhoseiny, T. El-Gaaly, and A. Elgammal. Digging deep into the layers of cnns: In search of how cnns achieve view invariance. *arXiv preprint arXiv:1508.01983*, 2015.

[2] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[3] A. Dosovitskiy and T. Brox. Inverting visual representations with convolutional networks. *arXiv preprint arXiv:1506.02753*, 2015.

[4] A. Fawzi and P. Frossard. Manitest: Are classifiers really invariant? In *British Machine Vision Conference (BMVC)*, pages 106.1–106.13, 2015.

[5] A. Fawzi, O. Fawzi, and P. Frossard. Analysis of classifiers' robustness to adversarial perturbations. *arXiv preprint arXiv:1502.02590*, 2015.

[6] O. Freifeld, S. Hauberg, K. Batmanghelich, and JW. Fisher III. Highly-expressive spaces of well-behaved transformations: Keeping it simple. In *International Conference on Computer Vision (ICCV)*, Santiago, Chile, December 2015.

[7] S. Gu and L. Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.

[8] M. Jaderberg, K. Simonyan, and A. Zisserman. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2008–2016, 2015.

[9] N. Karianakis, J. Dong, and S. Soatto. An empirical evaluation of current convolutional architectures' ability to manage nuisance location and scale variability. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[10] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1106–1114, 2012.

[11] K. Lenc and A. Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 991–999, 2015.

[12] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5188–5196, 2015.

[13] S. Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012.

[14] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6): 1087–1092, 1953.

[15] S-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] O.M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. *British Machine Vision (BMVC)*, 1(3):6, 2015.

[17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[18] S. Soatto and A. Chiuso. Visual representations: Defining properties and deep approximation. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[19] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.