Fine-grained Recognition in the Noisy Wild: Sensitivity Analysis of Convolutional Neural Networks Approaches

Erik Rodner[1]
erik.rodner@uni-jena.de
Marcel Simon[1]
marcel.simon@uni-jena.de
Robert B. Fisher[2]
rbf@inf.ed.ac.uk
Joachim Denzler[1]
joachim.denzler@uni-jena.de

[1] Computer Vision Group
Friedrich Schiller University Jena
Germany
www.inf-cv.uni-jena.de

[2] University of Edinburgh
United Kingdom

Abstract

In this paper, we study the sensitivity of CNN outputs with respect to image transformations and noise in the area of fine-grained recognition. In particular, we answer the following questions (1) how sensitive are CNNs with respect to image transformations encountered during wild image capture?; (2) how can we predict CNN sensitivity?; and (3) can we increase the robustness of CNNs with respect to image degradations? To answer the first question, we provide an extensive empirical sensitivity analysis of commonly used CNN architectures (AlexNet, VGG19, GoogleNet) across various types of image degradations. This allows for predicting CNN performance for new domains comprised by images of lower quality or captured from a different viewpoint. We also show how the sensitivity of CNN outputs can be predicted for single images. Furthermore, we demonstrate that input layer dropout or pre-filtering during test time only reduces CNN sensitivity for high levels of degradation.

Experiments for fine-grained recognition tasks reveal that VGG19 is more robust to severe image degradations than AlexNet and GoogleNet. However, small intensity noise can lead to dramatic changes in CNN performance even for VGG19.

1 Introduction

Convolutional neural networks (CNN) are currently the method of choice to model a map from visual data to semantic information in applications, such as image classification [14, 24, 26], video categorization [13], object detection [9], semantic segmentation [1, 8] and many more. In contrast to previous approaches with hand-designed feature extraction, CNN-based approaches learn relevant features in the form of convolutions directly from the given data.

The driving research question of our paper is how sensitive CNN outputs are to image noise and geometric transformations in the area of fine-grained recognition. While in nearly

© 2016. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

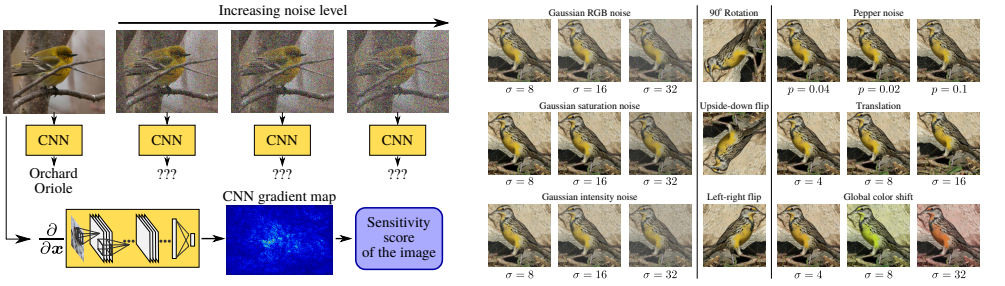

Figure 1: (Left) How sensitive are CNNs with respect to image noise and transformations? We study this question and show how to predict CNN sensitivity for a given image. (Right) Examples for the image degradations we use in our paper applied to a single image of the CUB-200-2011 dataset [31]. This figure is best viewed in color.

every computer vision paper, the evaluation is focused on the expected classification or segmentation accuracy, a more detailed analysis of the sensitivity of the results is missing or at least only restricted to the perturbations included in a fixed test set. However, being aware of weaknesses of different architectures is crucial to avoid unexpected behavior when the application is deployed. This especially applies to applications where the camera used in the field significantly differs from the camera used to record the training data [23]. Fine-grained recognition used for animal monitoring is one example. Everyday applications like smartphone apps which use the phone's camera is another example.

Figure 1 gives an overview of our paper. We provide a sensitivity analysis and prediction approach for common CNN architectures and fine-grained recognition. First, our analysis allows for selecting networks as well as raising awareness of instabilities that might occur when certain perturbations appear more frequently in a new application than in the birth place of nearly all pre-trained networks, the ImageNet challenge data [22]. Second, we show how to compute analytic sensitivity estimates for a given test image without explicitly altering the input image. Our approach is based on a first-order approximation of the CNN outputs and can be computed with a single backward pass through the network. Third, we show that including a dropout layer directly after the input can significantly boost the classification performance for high noise levels.

Our analysis can also be seen as stochastically studying the "zone of convergence" of convolutional neural networks, *i.e.*. the neighbourhood of an image that leads to the same prediction. Studying these properties of the CNN decision space is important to increase the robustness of visual inference algorithms.

Structure of the paper: The paper is structured as follows: We first review related work in the area of sensitivity analysis. An experimental evaluation of CNNs applied to transformed or noisy images is provided in Sect. 2. Sect. 3 shows how to increase the robustness of CNNs with respect to certain degradations and Sect. 4 derives a new algorithm that allows for predicting the sensitivity of a given image. The results of the paper are discussed and concluded in Sect. 5 and 6.

1.1 Related work

Related work on sensitivity analysis of classifiers The sensitivity of standard neural networks (multi-layer perceptrons, MLP) has been studied in the early nineties [2, 9] both with respect to weight and input perturbations. These works also relate the sensitivity to gradient estimates, but are restricted to both standard MLPs and simple vector representations rather than images as in our case. A very recent theoretical analysis only limited to linear and quadratic classifiers is given in [3]. Furthermore, the study of [3] focuses on adversarial noise rather than random noise.

Related work on sensitivity analysis of convolutional neural networks The sensitivity of classification systems is related to the concept of adversarial [30] and rubbish examples [17]. Adversarial examples are slightly modified images which show a significant change in the model output compared to the original image. The work of Szegedy [30] shows that these alterations can be computed by constrained gradient descent optimization. Furthermore, [6] presents the idea of adversarial training, where adversarial examples are added as additional training examples. Another strategy to increase the robustness with respect to adversarial examples is given by [7], where contractive networks are proposed that add an additional regularization during training to penalize large gradients with respect to the input data. This strategy has been already exploited for autoencoder training previously [20, 21]. The idea of rubbish examples [17] studies random noise images that lead to arbitrary classification decisions although their appearance can not be related to the particular object category or any natural image at all. In contrast to these studies of general robustness, we focus on sensitivity with respect to specific but very common classes of image degradations encountered during wild image capture.

The work of [15] studies equivariance (with invariance being a special case) and equivalence properties of CNNs by explicitly learning transformations of the output to compensate for given degradations. They show that although CNN outputs are not invariant to geometric transformations, they are approximately equivarient. The paper of [12] analyzes the marginalization properties of CNNs with respect to planar translations, scaling, and size of the context around a given bounding box. In contrast to these two works, we mainly focus our analysis on perturbations which occur during the image aquisition process like Gaussian and salt&pepper noise. Goodfellow *et al.* [5] analyzes the sensitivity in terms of a firing rate, which captures whether a neuron increased or decreased its value above or below a given threshold when pertubations are applied. While their analysis is well suited for analyzing the invariance of generic intermediate activations, it is difficult to interpret and understand the results in terms of classification accuracy, which is the task we are interested in.

Related work on data augmentation techniques A very natural method to decrease the sensitivity with respect to certain transformations is to perform explicit data augmentation by applying perturbations to the training images. The VGG19 model [26], for example, was trained using color shift data augmentation during training. An implicit data augmentation technique is given in [16], where an additional regularization term is used to minimize the Kullback Leibler divergence between the original and the adversarial posterior distribution. In contrast, we study data augmentation and pre-filtering as techniques to explicitly improve CNN performance in the presence of non-adversarial noise.

	AlexNet CUB200-2011		VGG CUB200-2011		GoogLeNet CUB200-2011		AlexNet Oxford Flowers		AlexNet Oxford Pets	
perturbation	$p_\triangle$	acc.	$p_\triangle$	acc.	$p_\triangle$	acc.	$p_\triangle$	acc.	$p_\triangle$	acc.
no perturbations	0.00%	62.56%	0.00%	81.29%	0.00%	77.75%	0.00%	86.96%	0.00%	79.59%
90° rotation	60.60%	32.22%	51.71%	45.93%	58.78%	38.32%	17.79%	78.06%	46.44%	48.49%
upside down flip	65.84%	27.06%	61.32%	36.18%	65.08%	31.72%	20.59%	75.87%	56.91%	39.17%
left right flip	14.05%	62.56%	10.01%	81.15%	9.87%	77.49%	6.18%	86.65%	9.89%	79.45%

Table 1: Label change probability $p_\triangle$ and recognition rates for some transformations on CUB-200-2011 and with different convolutional neural network architectures fine-tuned on the training set of CUB-200-2011.

2 How sensitive are CNN approaches?

In this section, we analyze the sensitivity of three state-of-the-art CNN architectures, which are widely used in recent works: AlexNet [14], VGG19 [26] and GoogLeNet [29]. We show in the experiments the weaknesses of a network that is trained on images which contain almost no noise. This is particularly important in real-world applications, where either low budget cameras are used or the lighting conditions changed after training.

Experimental setup We use models pre-trained on the ILSVRC2012 dataset provided by [11] and [26]. ILSVRC12 [22] is a large-scale dataset containing roughly 1.5 million training images split into 1000 object categories like car, person, cup, etc. Fine-tuning to the application-specific dataset is used, as this is a common step in most tasks. All experiments are performed using the CUB200-2011 birds dataset [31], which is one of the most commonly used datasets in fine-grained recognition. It contains 11788 images of 200 north American bird species. In addition, we also performed experiments on Oxford Flowers 102 [18] and the Oxford Pets dataset [19]. We use the split into training and test provided with the datasets.

The influence of noise is measured by the label change probability $p_\triangle$ and the classification accuracy *acc.*. The label change probability is the expected probability that the prediction of the CNN changes if random noise is applied to the image. In other words, if $p_\triangle$ is high, the class with the highest predicted probability is likely to change. The classification accuracy measures the performance for the classification task itself, which is related to the label flip probability of the initially correctly classified images.

All images are reduced to the given ground-truth bounding boxes, since we want to focus on transformations applied on the objects themselves and not on contextual background. Perturbations are applied to resized images fitting the input layer of the CNN. For translations, we translate the bounding box and crop it from the full image to avoid boundary effects.

Image noise and geometric transformations We compare several types of noise which occur in real-world applications. Figure 1 (right) shows differently degraded versions of an image illustrating the impact of the noise types and parameters on the visual appearance.

First, we consider random Gaussian noise affecting different aspects of the image. Gaussian RGB noise denotes random additive noise applied to each channel independently with mean 0 and standard deviation σ, while each pixel in a channel has a value in $[0,255]$. Gaussian intensity noise adds the noise to the intensity channel of the HSI image. Color shift and saturation noise work on the HSI image as well by adding Gaussian noise to the hue and saturation channel, respectively. Global color shift adds the same value to all pixels while local

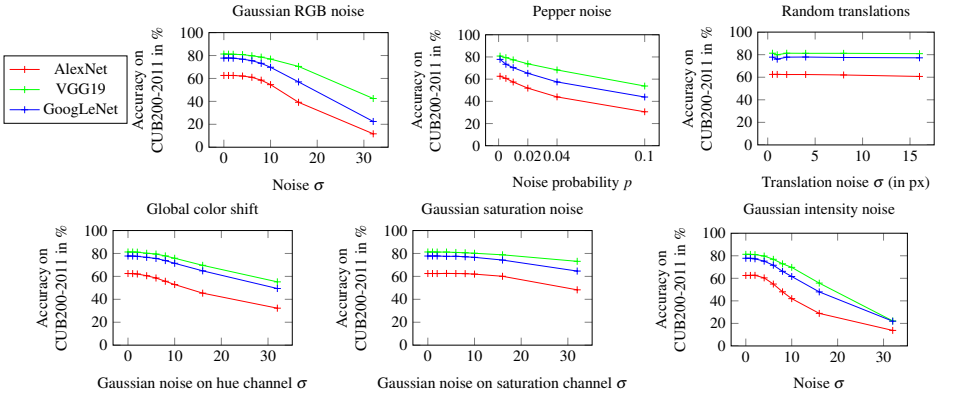

Figure 2: Comparison of classification accuracy on CUB200-2011 for Gaussian noise of different types as well as pepper noise and random translations.

color shift adds a different value to each pixel's hue channel. The second type of noise are geometric perturbations of the image. We consider random translations, rotations and flips. Finally, impulse noise is simulated by randomly setting pixel values to 0, independently for each channel. Hence, we call it pepper noise in the figures.

Evaluation The results for CUB200-2011 are given in Figure 2 for the random noise types as well as in Table 1 for the geometric perturbations. The baseline for the classification accuracy is measured on the original test images.

The rotation as well as the upside down flip cause a fairly high label change probability $p_{\triangle}$ and performance degradation. Interestingly, the classification accuracy is almost unchanged for left right flips and all random translations up to $\sigma = 16px$. This is indeed reasonable for our bird recognition task. Among random noise, the influence of pepper noise is drastic. Setting only 4% of the image pixels to intensity 0 causes a drop in classification performance of 20% for AlexNet and GoogLeNet as well as 10% for VGG19. This is surprising since the visual appearance is hardly effected for the human eye (see Figure 1, right) for this noise type. All Gaussian noise types have a strong influence on accuracy with intensity noise having the most and saturation noise the least influence.

Fig. 3 shows results for the Oxford Flowers 102 and the Oxford Pets dataset comparing the different noise types for AlexNet. In summary, similar conclusions compared to CUB200-2011 can be drawn. Gaussian intensity noise has by far the strongest impact on accuracy. A noise standard deviation of $\sigma = 16$ causes the recognition rate to drop by half to 43.6% for Oxford Flowers. Noise on the hue channel for every pixel and Gaussian RGB noise have slightly less influence, but are still very noticeable. Global color shift and noise on the saturation channel of the images have only little influence. The same is true for flips and rotations, which is reasonable due to flowers being close to rotationally symmetric.

3 Can we make CNNs more robust?

Our experiments showed that even small random noise can lead to a dramatic performance decrease. Now the question naturally arises whether it is possible to increase the robustness

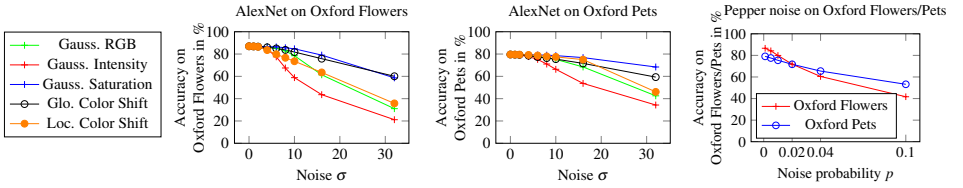

Figure 3: Comparison of classification accuracy on Oxford Flowers 102 and the Oxford Pets dataset for a fine-tuned convolutional neural network with AlexNet architecture.

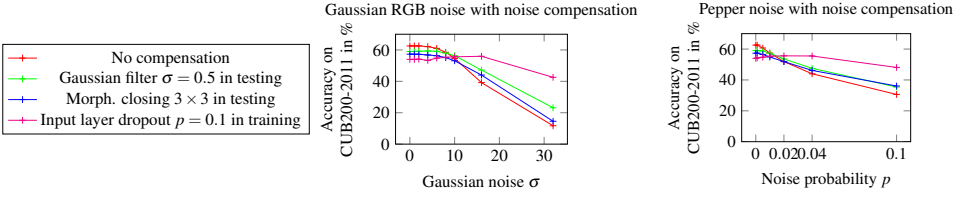

Figure 4: Can CNN sensitivity be reduced? We test Gaussian filtering and morphological closing of test images before CNN prediction as well as input layer dropout during training.

either during testing or by adapting the learning.

Robustness by test image denoising A first idea to handle noisy images is image denoising techniques. We evaluated whether simple linear and non-linear noise removal approaches are suitable to reduce the sensitivity of CNNs. Figure 4 shows the results for AlexNet in combination with Gaussian filtering and morphological closing on CUB200-2011.

Filtering the input images with a Gaussian filter of size 3×3 and $\sigma = 0.5$ improved the performance if the variance of the Gaussian RGB noise is greater than 10 or the pepper noise is more frequent than $p = 2\%$. Using a larger Gaussian filter decreases the accuracy significantly and performs worse even though it leads to increased robustness. The morphological closing operations slightly improves accuracy in case of strong noise, however, the accuracy is worse than a Gaussian filtering with $\sigma = 0.5$.

The results reveal that preprocessing improves the robustness, but decreases the accuracy for images without any noise. For noisy test images, the accuracy is higher if noise removal is only subtle. This result is intuitive as noise removal can lead to a loss in information.

Robustness by augmented training Instead of treating noise at test time, invariance to noise can be also learned at training time. Specifically, random noise can be added to the training data during CNN learning. We used a dropout layer [28] added between the input data and the first convolutional layer of the CNN. Dropout indirectly augments the training set by setting neuron outputs to zero at random, which is in our case equivalent of adding noise directly to the images during training. In our case, the dropout probability was set to 0.1 and the results are shown in Figure 4.

Similar to preprocessing the input image, augmented training of a CNN reduces its performance on noise-free test images. However, the performance on noisy images is greatly improved and the CNN is more robust to strong random noise. The results suggest that augmented training is beneficial if the test domain is indeed characterized by high degrees of

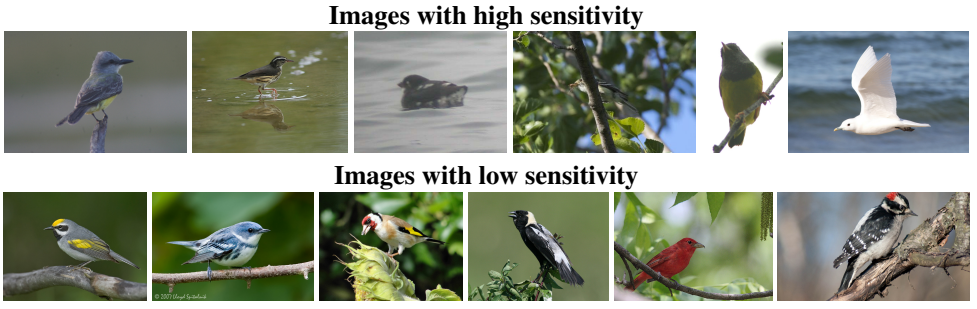

Figure 5: Sample images with a high or low standard deviation of the classification score with respect to Gaussian noise ($\sigma = 1$). Images have been cropped to ground-truth bounding boxes and scaled according to the CNN input layer.

image noise compared to the training domain. In any other case, focusing on the training images without any noise augmentation seems to be the better choice.

Qualitative results The results demonstrate that a large portion of the class predictions change even if only a small amount of noise is added. Fig. 5 shows sample images whose classification score significantly or only marginally changed when noise is added. High sensitivity images are classified by the CNN with high confidence if the image is free of noise, but their maximum classification score changes significantly when even small noise is added (Gaussian RGB noise with $\sigma = 1$). Low sensitivity images behave in the opposite way.

As can be seen in Fig. 5, high sensitivity images are characterized by low contrast objects, whereas low sensitivity images often have discriminative color patterns. To validate this observation, we computed the entropy of hue values within the bird region for the 100 least sensitive and the 100 most sensitive images. Whereas the least sensitive images had a mean hue entropy of 6.12, the most sensitive images were characterized by a mean hue entropy of 5.51. An additional Wilcoxon rank sum test also showed that this difference in mean values is indeed significant ($p < 10^{-10}$) and our observation for Fig. 5 is valid.

4 Can we predict CNN sensitivity for a test image?

Since we now know that CNN outputs can be sensitive to certain transformations and noise processes, the question remains whether we can quickly detect images with unstable CNN outputs. This question goes beyond a pure sensitivity study but asks for uncertainty estimates often available for Bayesian methods but not for CNNs. In the following, we derive a method for estimating sensitivity scores that does not require a costly explicit alteration of the image.

Predicting sensitivity with a backward pass Let $\boldsymbol{f}(\boldsymbol{z}) \in \mathbb{R}$ be the single output of a CNN for an image $\boldsymbol{z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_C) \in \mathbb{R}^{D \cdot C}$ represented as a flattened vector with D pixels and C channels. In the following, we assume that the input image is altered by $\boldsymbol{g}(\boldsymbol{z}, \boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^N$ is a random variable controlling the perturbation and without loss of generality, we assume that $\boldsymbol{g}(\boldsymbol{z}, \boldsymbol{0}) = \boldsymbol{z}$. We are now considering the change $\triangle \boldsymbol{f}$ of CNN outputs when

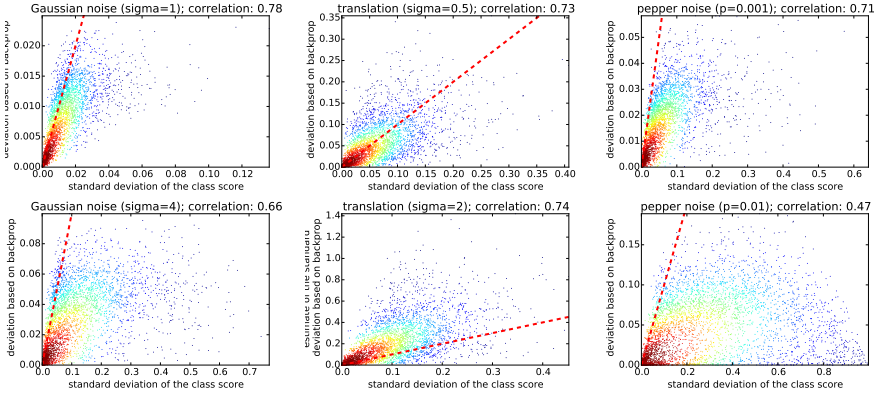

Figure 6: Scatter density plots of our sensitivity prediction for (Left) Gaussian RGB noise with $\sigma \in \{1,4\}$, (Center) translations with $\sigma \in \{0.5,2\}$, and (Right) pepper noise with $p \in \{0.001,0.01\}$. Each point in the scatter plot corresponds to a single image in the CUB dataset, which we transformed with a given perturbation 10 times. The colors indicate the density in the plot. The diagonal red dashed line corresponds to the identity of the estimate from gradients and the empirical estimate from different trials.

applying $\boldsymbol{g}$ to an image $\boldsymbol{z}'$ and using a first-order approximation:

$$\triangle \boldsymbol{f} = \boldsymbol{f}(\boldsymbol{g}(\boldsymbol{z}', \boldsymbol{\theta}')) - \boldsymbol{f}(\boldsymbol{g}(\boldsymbol{z}', \boldsymbol{0})) \approx \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}^T |_{\boldsymbol{z}=\boldsymbol{z}'} \cdot \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\theta}}^T |_{\boldsymbol{\theta}=\boldsymbol{0}} \cdot \boldsymbol{\theta}' = \boldsymbol{F}^T \cdot \boldsymbol{G}^T \cdot \boldsymbol{\theta}' \tag{1}$$

The matrix $\boldsymbol{G} \in \mathbb{R}^{N \times C \cdot D}$ is the Jacobian of the perturbation evaluated at $\boldsymbol{\theta} = 0$. The gradients $\boldsymbol{F} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}} \in \mathbb{R}^{C \cdot D}$ can be easily computed with a backward pass [25, 27]. They have been used by previous work both for computing saliency and segmentation [27] as well as for part discovery [25]. The above result is quite intuitive also from the perspective of saliency maps. If the gradient map shows high saliency values throughout the whole image, the influence of Gaussian RGB noise on the CNN output is likely to be high. This is not the case if only a small part of the image is occupied by the object achieving the highest saliency. The transformation $\boldsymbol{g}$ can be viewed as a stochastic process and we derive an approximation for the variance of the CNN output as follows, where we assume that the expectation with respect to $\boldsymbol{\theta}$ is the output for the original image $\boldsymbol{z}'$:

$$\begin{aligned} \mathbb{V}_{\boldsymbol{\theta}}(\boldsymbol{z}') &= \mathbb{E}_{\boldsymbol{\theta}} \left(\boldsymbol{f}(\boldsymbol{g}(\boldsymbol{z}', \boldsymbol{\theta}) - \boldsymbol{f}(\boldsymbol{z}')) \right)^2 \approx \mathbb{E}_{\boldsymbol{\theta}} \left(\boldsymbol{\theta}^T \boldsymbol{G} \boldsymbol{F} \boldsymbol{F}^T \boldsymbol{G}^T \boldsymbol{\theta} \right) \\ &= \mathrm{tr} \left(\mathbb{E}_{\boldsymbol{\theta}} (\boldsymbol{\theta} \boldsymbol{\theta}^T) \cdot \boldsymbol{G} \boldsymbol{F} \boldsymbol{F}^T \boldsymbol{G}^T \right) \end{aligned} \tag{2}$$

We use tr() to denote the trace of a given matrix. In the following paragraphs, we study different types of perturbations and their resulting gradients $\boldsymbol{G}$. We refer to the variance in the above formula as the *sensitivity score*.

Predicting iid. image noise sensitivity A simple model for perturbations is additive iid. Gaussian (RGB) noise on the image, which we already analyzed empirically in Sect. 2: $g^{\text{gauss}}(\boldsymbol{z}, \boldsymbol{\theta}) = \boldsymbol{z} + \boldsymbol{\theta}, \quad \boldsymbol{\theta} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$, where $\boldsymbol{\theta}$ is an $C \cdot D$-dimensional random vector. The

Jacobian of the transformation is simply the identity matrix, *i.e.*. $\boldsymbol{G} = \boldsymbol{I} \in \mathbb{R}^{C \cdot D \times C \cdot D}$ and we obtain: $\mathbb{V}_{\boldsymbol{\theta}}^{\text{gauss}}(\boldsymbol{z}') = \sigma^2 \|\boldsymbol{F}\|_2^2$. Our sensitivity score is therefore proportional to the magnitude of the gradient map.

Let us now consider pepper noise that affects every (RGB) pixel of the image by setting them to black pixels with probability p. Following our notation this can be expressed by: $g^{\text{pepper}}(\boldsymbol{z}, \boldsymbol{\theta}) = [\boldsymbol{z}_1 * (1 - \boldsymbol{\theta}), \ldots, \boldsymbol{z}_C * (1 - \boldsymbol{\theta})]$, where $\boldsymbol{\theta} \in \{0,1\}^D$ is a vector of independent Bernoulli random variables, and $*$ denotes component-wise multiplication. The Jacobian $\boldsymbol{G}$ is a $\mathbb{R}^{D \times C \cdot D}$ matrix and the multiplication with $\boldsymbol{F}$ gives us

$$\boldsymbol{G}\boldsymbol{F} = \Big[-\sum_{c=1}^{C} z_{k+D\cdot(c-1)} \cdot F_{k+D\cdot(c-1)} \Big]_{k=1}^{D} = \boldsymbol{v} \in \mathbb{R}^D, \tag{3}$$

a vector of length D, which is the number of pixels D of image $\boldsymbol{z}$. The matrix $\mathbb{E}_{\boldsymbol{\theta}}(\boldsymbol{\theta} \cdot \boldsymbol{\theta}^T)$ depends on the noise probability $0 \leq p \leq 1$: $\mathbb{E}_{\boldsymbol{\theta}}(\boldsymbol{\theta} \cdot \boldsymbol{\theta}^T) = p^2 \boldsymbol{e}\boldsymbol{e}^T - p(p-1)\boldsymbol{I}$, with $\boldsymbol{e} = (1, \ldots, 1)^T$. Finally after combining, we have:

$$\mathbb{V}_{\boldsymbol{\theta}}^{\text{pepper}}(\boldsymbol{z}') = p^2 (\boldsymbol{v}^T \boldsymbol{e})^2 - p(p-1)\|\boldsymbol{v}\|^2. \tag{4}$$

Predicting image translation sensitivity Following our analysis in Sect. 2, the gradients with respect to small image translations $\boldsymbol{\theta} \in \mathbb{R}^2$ for all channels are

$$\boldsymbol{G} = \begin{bmatrix} \frac{\partial \boldsymbol{z}_1}{\partial x} & \cdots\cdots & \frac{\partial \boldsymbol{z}_{D\cdot C}}{\partial x} \\ \frac{\partial \boldsymbol{z}_1}{\partial y} & \cdots\cdots & \frac{\partial \boldsymbol{z}_{D\cdot C}}{\partial y} \end{bmatrix} \tag{5}$$

and depend on the image gradients in each channel. Since we consider a Gaussian model for the translations $\boldsymbol{\theta} \in \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$, we end up with: $\mathbb{V}_{\boldsymbol{\theta}}^{\text{trans}} = \sigma^2 \|\boldsymbol{G}\boldsymbol{F}\|_2^2$. where image gradients are multiplied with the CNN gradient map. This result is quite intuitive since large image gradients corresponding to edges will likely lead to a high sensitivity of the CNN with respect to small translations.

Validating our sensitivity prediction How accurate is our sensitivity prediction? To answer this question, we correlate our approximations for $\mathbb{V}$ with the actual empirical standard deviation of the output change and provide scatter plots in Figure 6. A single point in the scatter plot corresponds to a single perturbation of one of the CUB test images colored with its density value in the plot to improve visualization of the distribution of data points. CNN outputs are computed using AlexNet.

As can be seen, the sensitivity prediction is quite accurate for small perturbations with a high correlation (given in the title of each figure), which is reasonable since our method is based on a linear approximation. For pepper and Gaussian RGB noise, however, large perturbations lead to a smaller correlation of our sensitivity score with our empirical estimates. This is due to our linear approximation with gradient estimates, which is only valid for smaller noise levels, and the small number of perturbed samples (10 in our case) we use for our empirical estimates.

5 Discussion

The experiments show that the influence especially of common intensity noise is severe even at low noise levels. The reason is a domain shift between noise-free training and pertubated test data. From our study, we can draw several conclusions:

1. The training images should have the same noise level as the test images and care has to be taken even for small noise applied to intensities.
2. Data augmentation during training is not the solution as it decreases the accuracy on noise free images dramatically and is only beneficial for high noise levels as shown.
3. Noise sensitivity depends on the CNN architecture and VGG19 has shown to be the most robust one.
4. Sensitivity of CNN outputs can be predicted for small noise levels with our technique in Sect. 4 allowing for uncertainty estimates of CNN outputs.

These conclusions can be seen as guidelines especially for developers of real-world applications, where, for example, cheap camera sensors deliver low quality images but the training was performed on relatively noise-free datasets like ImageNet. We studied sensor-related perturbations on purpose, since they are often neglected in the era of huge datasets and benchmarks. Simulating perturbations might not be relevant in scenarios where the sensor used for acquisition during training is equivalent to the one during testing. In this case, a larger dataset already contains examples of realistic noise, such as different viewpoints and object deformations. However, our evaluations are important when it comes to the change to or the use of low-cost sensors with higher noise levels and wrong color calibration.

As it is impossible to simulate all possible real-world perturbations, we only consider a subset of them. Missing are especially geometric distortions including radial distortion as well as white-balance, which are hard to realistically simulate. The results for translation and global color shift, respectively, lead to the assumption that their influence is fairly low. Second, multiple kinds of noise occur in combination. As the addition of noise is likely to decrease accuracy, the combination of multiple noise types might lead to an even stronger degradation. For example, Gaussian intensity noise with $\sigma = 4$ combined with pepper noise with probability $p = 0.02$ yields 48.5% accuracy on CUB200-2011. Compared to only Gaussian intensity noise, the accuracy is almost 12% lower and more than 3% lower compared to only applying pepper noise.

6 Conclusions

In this paper, the sensitivity of common CNN architectures is analyzed empirically and analytically. The experiments reveal that especially the most common AlexNet architecture is very sensitive to strong non-adversarial random noise leading to a significant drop in classification performance. VGG19 is more robust in terms of stability with respect to Gaussian and pepper noise as well as other non-random perturbations. Nevertheless, the effect of noise with medium strength already has a significant impact on accuracy in fine-grained recognition tasks. Two common approaches for dealing with noise were evaluated and turned out to improve stability only for highly degraded images in the test domain. We further provide an estimation of the sensitivity of a CNN for a given image and show that its predictions strongly correlate with our expensive empirical estimates.

In future work, we plan to study the relationship between architectural choices in a CNN and its sensitivity to perturbations in more detail, *e.g.* with ResNet-like architectures [10] with multiple depths.

Acknowledgements Part of this research was supported by grant RO 5093/1-1 of the German Research Foundation (DFG).

References

[1] Clemens-Alexander Brust, Sven Sickert, Marcel Simon, Erik Rodner, and Joachim Denzler. Convolutional patch networks with spatial prior for road detection and urban scene understanding. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 510–517, 2015.

[2] Jin Young Choi and Chong-Ho Choi. Sensitivity analysis of multilayer perceptron with differentiable activation functions. *IEEE Transactions on Neural Networks*, 3(1): 101–107, 1992.

[3] Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Fundamental limits on adversarial robustness. In *ICML Deep Learning Workshop*, 2015. arXiv:1502.02590.

[4] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jagannath Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 580–587. IEEE, 2014.

[5] Ian Goodfellow, Honglak Lee, Quoc V Le, Andrew Saxe, and Andrew Y Ng. Measuring invariances in deep networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 646–654, 2009.

[6] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015. arXiv:1412.6572.

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

[8] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *Computer Vision and Pattern Recognition (CVPR)*, pages 447–456, 2015.

[9] Sherif Hashem. Sensitivity analysis for feedforward artificial neural networks with differentiable activation functions. In *International Joint Conference on Neural Networks (IJCNN)*, volume 1, pages 419–424. IEEE, 1992.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[11] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross B Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia*, volume 2, page 4, 2014.

[12] Nikolaos Karianakis, Jingming Dong, and Stefano Soatto. How well can a cnn marginalize simple nuisances it is designed for? *arXiv preprint arXiv:1505.06795*, 2015.

[13] Andrej Karpathy, George Toderici, Sachin Shetty, Tommy Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1732. IEEE, 2014.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[15] Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. *arXiv preprint arXiv:1411.5908*, 2014.

[16] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. Distributional smoothing by virtual adversarial examples. *arXiv preprint arXiv:1507.00677*, 2015.

[17] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *arXiv preprint arXiv:1412.1897*, 2014.

[18] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008.

[19] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, pages 3498–3505, 2012.

[20] Salah Rifai, Grégoire Mesnil, Pascal Vincent, Xavier Muller, Yoshua Bengio, Yann Dauphin, and Xavier Glorot. Higher order contractive auto-encoder. In *Machine Learning and Knowledge Discovery in Databases*, pages 645–660. Springer, 2011.

[21] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *International Conference on Machine Learning (ICML)*, pages 833–840, 2011.

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, pages 1–42, 2014.

[23] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, pages 213–226, 2010.

[24] Marcel Simon and Erik Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *International Conference on Computer Vision (ICCV)*, pages 1143–1151, 2015.

[25] Marcel Simon, Erik Rodner, and Joachim Denzler. Part detector discovery in deep convolutional neural networks. In *Asian Conference on Computer Vision (ACCV)*, volume 2, pages 162–177, 2014.

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[27] Katen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *International Conference on Learning Representations (ICLR)*, 2014.

[28] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[29] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[30] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014. arXiv:1312.6199.

[31] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.