Deep Convolutional Neural Networks and Noisy Images

Tiago S. Nazaré, Gabriel B. Paranhos da Costa, Welinton A. Contato, and Moacir Ponti

Instituto de Ciências Matemáticas e de Computação — Universidade de São Paulo
13566-590 São Carlos, SP, Brazil
[tiagosn,gbpcosta,welintonandrey,ponti]@usp.br

Abstract. The presence of noise represent a relevant issue in image feature extraction and classification. In deep learning, representation is learned directly from the data and, therefore, the classification model is influenced by the quality of the input. However, the ability of deep convolutional neural networks to deal with images that have a different quality when compare to those used to train the network is still to be fully understood. In this paper, we evaluate the generalization of models learned by different networks using noisy images. Our results show that noise cause the classification problem to become harder. However, when image quality is prone to variations after deployment, it might be advantageous to employ models learned using noisy data.

1 Introduction

In real-world applications image quality may vary drastically depending on factors such as the capture sensor used and lighting conditions. These scenarios need to be taken into consideration when performing image classification, since quality shift directly influence its results.

Lately, deep convolutional neural networks have obtained outstanding results in image classification problems. Nonetheless, little has been done to understand the impacts of image quality on the classification results of such networks. In most studies, networks were only tested on images whose quality is similar to the training set (i.e. similar noise/blur levels). The lack of research in this topic is not exclusive to deep learning applications. Most image classification systems neglect preprocessing [12] and assume that image quality does not vary [5].

Given that in real-world applications image quality may vary, we evaluate classification performance of classic deep convolutional neural networks when dealing with different types and levels of noise. In addition, we investigate if denoising methods can help mitigate this problem.

1.1 Related Work

We devote our efforts to investigate the effects of noisy images when using deep convolutional neural networks in classification tasks. There are papers investigating the effect of label noise in the learning capability and performance of

convolutional neural networks [2]. However, this problem is not addressed in this paper. Thus, in our experiments, we assume that all labels are correct.

Some studies have already identified that image quality can hinder classification performance in systems that employ neural networks [5] and in systems that use hand-crafted features [9][4]. Recently, the development of noise-robust neural networks has been investigated. For instance, [6] presented a network architecture that can cope with some types of nosy images, while [13] designed a network that is capable of dealing with noise in speech recognition.

Dodge and Karam [5] showed that state-of-the-art deep neural networks are affected when classifying images with lower quality. In their experiments, each network was trained on images from the original dataset (with a negligible amount of noise due to the image formation process) and, then, used to classify images from the same dataset on their original state, degraded by noise and affected by blur. Their results show that classification performance is hampered when classifying images with lower quality. However, their experiments do not cover the presence of low-quality images in the training set and their impact in the learned model.

Paranhos da Costa et al. [4] extended the methodology of [5] by considering that low-quality images can also appear in the training set. In their setup, several noisy versions of a dataset are created: each version has the same images as the original dataset, wherein all images are affected by a type of noise at a fixed level. They also evaluated the effects of denoising techniques by studying restoration of noisy images. Hand-crafted features (LBP and HOG) were extracted, SVM classifiers trained with each version of the training set and, then, used to classify all versions of the test set. Even so, their study only considered two hand-crafted features (LBP and HOG).

We believe that noise makes classification more difficult due due to the fact that models trained with a particular noisy/restored training set version – and tested on images with the same noise configuration – usually perform worse than a model trained and tested on the original data. Our empirical evaluation is based on [4], however there are two main differences. First, our experiments target deep neural networks, with the ability to learn from data, even from low quality data, while the previous study considers hand-crafted features and SVM classifiers. Second, we investigate if training models with a specific noise or image restoration setup can help to build models that are more resilient to changes in image quality for future data, i.e., in the test set.

2 Experiments

2.1 Experimental setup

The **first** step in the experimental setup used in this study is to create noisy and restored versions of each one of the three publicly-available datasets selected for our experiments: MNIST, CIFAR-10 and SVHN (further information on these datasets is presented in Section 2.2). To do that, five copies of

the original dataset are degraded by a Gaussian noise with standard deviation $\sigma = \{10, 20, 30, 40, 50\}$, and another five copies were hindered by a salt & pepper noise using $p = \{0.1, 0.2, 0.3, 0.4, 0.5\}$, where p is the probability of a pixel being affected by the noise. Next, denoising methods are applied on each of these versions, generating 10 restored versions of each dataset, one for each noisy version.

The restored versions of the datasets affected by Gaussian noise are obtained by filtering the images with the Non-Local Means (NLM) algorithm [3]. To perform the NLM denoising, we used a 7×7 patch, a 11×11 window and a set the parameter h equal to the standard deviation of the Gaussian noise used to corrupt the dataset being restored. Regarding the salt & pepper noise, all restored versions were generated by filtering the noise images using a 3×3 median filter. Hence, we have 21 different versions of each dataset (the original dataset, 10 noisy and 10 restored versions). Since all versions contain the same images, we always use the same training-test split presented in the original dataset paper.

The **second** step is to learn a classifier for each training set version. This means that a single network architecture was trained with each dataset version, creating 21 different classifiers. Then, each classifier was tested on all versions of the test set. This process is illustrated by the diagram shown in Figure 1. For the MNIST dataset, we selected an architecture similar to the LeNet-5 [10], while for the CIFAR-10 and the SVHN datasets, an architecture similar to the base model C of [15] was used. These architectures were implemented using the Keras library and our implementation was based on the code available on [1]. A convolutional neural network architecture was selected for each dataset.

In the **third** part of our experimental setup, we compare the learned models. We begin our analysis by comparing classification accuracies when both the training and test set have the same type and level of noise. By doing so, we want to measure how much harder classifying these datasets gets – for that network architecture – once the noise occurs at a particular level in the entire dataset (training and test set).

Additionally, we compare the results on the noisy versions with their restored counterparts, which allows us to measure how much the use of denoising techniques can help to improve accuracy. After, we visualize the classification results of all trained models in all versions of the test set using heatmaps. Such visualization shows how performance varies for each model. Lastly, we compute the mean (and standard deviation) of the accuracies obtained by each classifier over all test set versions. Using these values we can quantize the overall performance difference among models and compare models with regards to their resilience to different types of noisy images. This setup is illustrated by the diagram shown in Figure 1.

2.2 Datasets

MNIST: handwritten digits [10] broadly used in deep learning experiments due to being real-world data that requires minimal pre-processing/formatting.

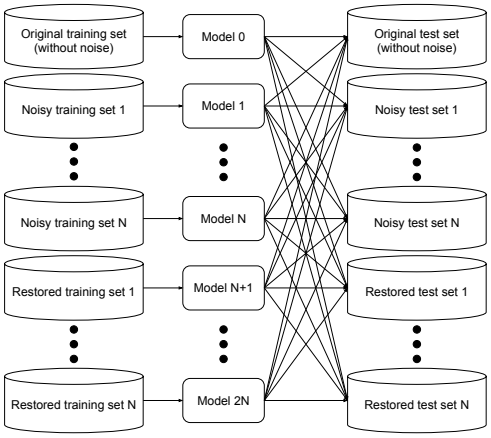

Fig. 1. Experimental setup diagram. In our experiments a different model is trained for each noise configuration, then these models are used to classify all versions of the test set. This figure was based on Figure 2 of [4].

CIFAR-10: consisting of 60,000 color 32×32 images equally split into 10 classes [8]. This dataset is subdivided into training and test sets, which include 50,000 and 10,000 images, respectively.

SVHN: house numbers from Google Street View images [11], defines a real-world problem of recognizing digits in natural images. It is composed by 73,257 images in the training set and 26,032 images in the test set.

3 Results and Discussion

To contrast the impact of noise and denoising methods in image quality, the average Peak Signal-to-Noise Ratio (PSNR) values for each dataset version are shown in Table 1. By comparing the results when training and testing is performed in the same dataset version it is possible to analyse how noise affects classification, in particular if it makes the task more difficult by changing the parameter space learned by the network. These results are shown in Table 2, in which it is possible to notice that, in general, the presence of noise, even when restored using a denoising algorithm, increases the complexity of the classification task.

To better understand the effects of using denoising methods we plot some of the results presented in Table 2 in Figure 3, showing the accuracies of the models trained with images affected by different levels of Gaussian noise as well salt & pepper noise, and their restored counterparts. Neither of these two figures present the results for the MNIST dataset, because, as can be seen in Table 2, the differences for this dataset were too small.

Despite the increase in PSNR when employing NLM for denoising, accuracy decreased when classifying data restored by this method. This is probably due

Table 1. Average PSNR for each noise level.

Noise		MNIST		CIFAR-10		SVHN	
		noisy	restored	noisy	restored	noisy	restored
Gaussian	$\sigma = 10$	30.93	33.18	28.25	29.88	28.15	32.65
	$\sigma = 20$	24.86	27.46	22.36	25.54	22.21	28.15
	$\sigma = 30$	21.35	24.35	18.99	22.81	18.81	25.21
	$\sigma = 40$	18.87	21.54	16.67	20.85	16.47	23.40
	$\sigma = 50$	16.96	17.84	14.95	19.49	14.74	22.26
s&p	$p = 0.1$	13.44	20.78	15.23	25.89	15.52	34.30
	$p = 0.2$	10.79	18.54	12.50	24.00	12.79	29.32
	$p = 0.3$	9.36	16.69	10.98	21.64	11.28	24.82
	$p = 0.4$	8.43	15.13	9.97	19.22	10.28	21.28
	$p = 0.5$	7.78	13.84	9.23	17.04	9.55	18.53

Table 2. Accuracy of each network when training and testing was conducted using the same dataset version.

Noise Type		MNIST		CIFAR-10		SVHN	
		noisy	restored	noisy	restored	noisy	restored
original		0.9903		0.8192		0.9438	
Gaussian	$\sigma = 10$	0.9916	0.9885	0.8007	0.8016	0.9415	0.9445
	$\sigma = 20$	0.9875	0.9878	0.7727	0.7583	0.9284	0.9210
	$\sigma = 30$	0.9898	0.9867	0.7309	0.7156	0.9015	0.8793
	$\sigma = 40$	0.9890	0.9851	0.6991	0.6625	0.8849	0.8455
	$\sigma = 50$	0.9860	0.9814	0.6608	0.6277	0.8542	0.7987
s&p	$p = 0.1$	0.9799	0.9861	0.7227	0.7613	0.9308	0.9418
	$p = 0.2$	0.9793	0.9802	0.6902	0.7508	0.9193	0.9425
	$p = 0.3$	0.9753	0.9718	0.6088	0.7138	0.9095	0.9301
	$p = 0.4$	0.9641	0.9605	0.5610	0.6921	0.9057	0.9239
	$p = 0.5$	0.9437	0.9426	0.5398	0.6627	0.8889	0.9228

to that fact such denoising procedure generate blurry images, removing relevant information, as can be seen in Figure 2.

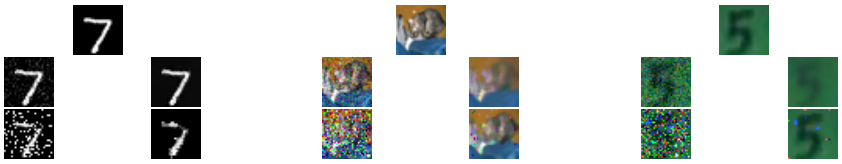

Fig. 2. Examples of noisy images for each dataset. The first row show the original images. The second row depict images with Gaussian noise ($\sigma = 30$) and their restored versions. Finally, the third row has images affected by salt & pepper noise ($p = 0.3$) and denoised by a median filter.

Next, results comparing models when classifying all dataset versions are presented using the heatmaps of Figure 4. Each row in a heatmap represent a version of the training set, while each column displays the results for a version of the test set. As demonstrated in [4], models tend to achieve their best accuracy when classifying data that has the same quality as the data used to train them. Nevertheless, depending on the training set, different generalization capability is achieved. This is demonstrated by models whose results are similar to their

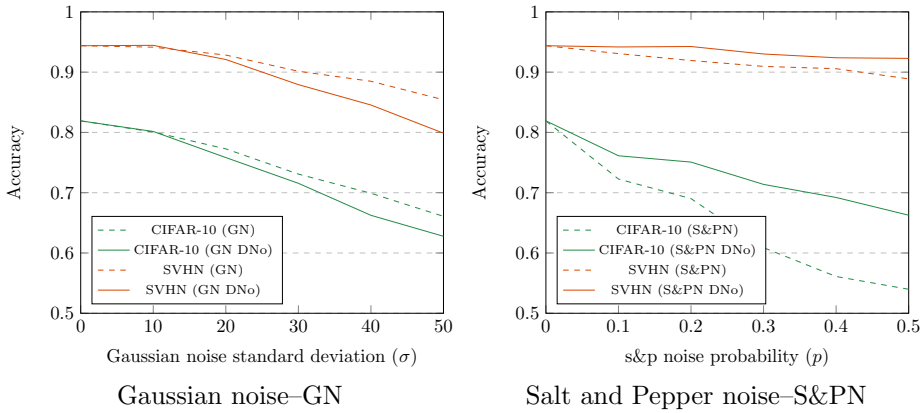

Fig. 3. Comparison of the accuracy of each network for different noise parameters (a) Gaussian noise standard deviation σ (b) Salt & Pepper probability p, with and without the use of denoising algorithms (DNo) for restoration.

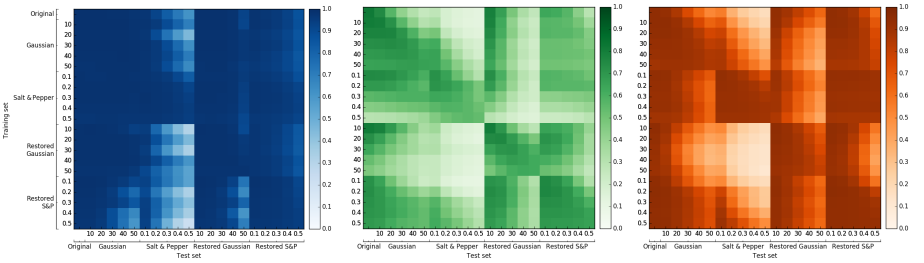

Fig. 4. Heatmaps representing the results obtained on MNIST (left), CIFAR-10 (center) and SVHN (right).

Table 3. Average accuracy and standard deviation (in percentages) for each model in all test set versions.

Noise Type		**MNIST**		**CIFAR-10**		**SVHN**	
		noisy	restored	noisy	restored	noisy	restored
original		93.13 ± 9.68		43.50 ± 21.84		68.21 ± 24.64	
Gaussian	$\sigma = 10$	94.59 ± 7.35	87.68 ± 18.68	45.04 ± 21.80	48.83 ± 21.12	71.87 ± 22.47	67.41 ± 25.15
	$\sigma = 20$	91.00 ± 14.08	90.77 ± 14.18	49.52 ± 20.41	48.68 ± 21.06	73.72 ± 20.28	67.41 ± 25.15
	$\sigma = 30$	94.26 ± 8.65	88.29 ± 18.43	50.72 ± 18.57	47.75 ± 18.48	74.36 ± 18.94	61.09 ± 28.32
	$\sigma = 40$	94.49 ± 6.94	92.95 ± 9.90	51.76 ± 16.79	41.82 ± 18.07	75.62 ± 17.54	61.01 ± 28.67
	$\sigma = 50$	93.58 ± 8.16	90.72 ± 12.42	51.23 ± 14.85	38.31 ± 16.04	75.18 ± 16.76	63.51 ± 27.81
s&p	$p = 0.1$	95.85 ± 4.02	90.12 ± 12.55	56.78 ± 15.04	44.20 ± 22.47	82.97 ± 11.81	66.69 ± 24.95
	$p = 0.2$	97.13 ± 2.12	84.71 ± 19.06	56.92 ± 12.89	51.69 ± 20.08	$\mathbf{83.41 \pm 11.47}$	76.33 ± 20.13
	$p = 0.3$	$\mathbf{97.27 \pm 1.74}$	88.13 ± 15.61	49.79 ± 11.89	57.20 ± 14.97	82.26 ± 13.01	77.27 ± 19.79
	$p = 0.4$	97.11 ± 1.57	83.70 ± 18.57	42.83 ± 11.85	$\mathbf{57.83 \pm 13.71}$	82.84 ± 13.28	80.14 ± 16.04
	$p = 0.5$	96.32 ± 1.87	81.66 ± 22.40	33.21 ± 12.21	57.79 ± 11.76	80.02 ± 14.14	82.16 ± 12.89

best, even when classifying data affected by other types of noise. To compare the noise resilience of these models, Table 3 show the mean and standard deviation

accuracies obtained by each classifier over all test set versions (the mean and standard deviation of each row of each heatmap).

In this comparison, the network trained with the original dataset is used as a baseline scenario, given that this network has no previous knowledge of any type of noise, while the others have already seen noisy images in some level. Therefore, networks trained with noisy images have an advantage when dealing with noise in future data even when it occurs at a different level. For the MNIST dataset, models trained on the original data obtained an average accuracy of 93.13% and a standard deviation of 9.68%, while the best overall result of $97.27 \pm 1.74\%$ was obtained by the model trained using images affected by salt & pepper noise (with $p = 0.3$). On the CIFAR-10 dataset, the best average results ($57.83 \pm 13.71\%$) were obtained by the model trained with data corrupted by salt & pepper with $p = 0.4$ and restored using the median filter. The model trained using the original data obtained $43.50 \pm 21.84\%$. Lastly, in the SVHN dataset, the model trained on the original dataset obtained a overall $68.21 \pm 24.64\%$ accuracy, against $83.41 \pm 11.47\%$ obtained by the model trained with images affected by salt & pepper noise ($p = 0.2$).

Nevertheless, it is possible to notice that some models are better at generalising to other types of noise. For instance, in the MNIST dataset, most models trained with salt & pepper noise obtained were able to achieve results around 0.6 or higher, while the other model did not.

To facilitate reproducibility of our experiments, our code is publicly available at `http://github.com/tiagosn/dnnnoise2017`.

4 Conclusions

We analysed the behaviour of deep convolutional neural networks when dealing with different types of image quality. Our study covered images affected by s&p and Gaussian noise and their restored versions. Although noise injection in the training data is a common practice, our systematic methodology provide a better understanding of the behaviour of the models under noise conditions. The results indicate that training networks using data affected by some types of noise could be beneficial for applications that need to deal with images with varying quality, given that it seems to improve the resilience of the network to other types of noise and noise levels.

Concerning denoising methods, images restored with the median filter, when compared against images with s&p noise, were able to improve the accuracy in data with the same quality. Nevertheless, models trained with s&p noise usually obtained a better noise resilience. Restoring images with NLM resulted, for the most part, in a decrease in performance. This was probably due to the removal of relevant information caused by NLM smoothing. Hence, better results might be achieved with a different parameter choice.

As future work we intend to explore deeper models such as VGG [14] and ResNets [7]. These experiments should also include neural networks designed to

be robust to noise such as in [6]. Moreover, we aim at conducting experiments in larger datasets like ImageNet.

5 Acknowledgment

The authors would like to thank FAPESP (grants #16/16111-4, #13/07375-0, #15/05310-3, #15/04883-0)

References

1. Arnold, T.: Stat 365/665: Data mining and machine learning: Lecture notes (transfer learning and computer vision I) (April 2016)
2. Bekker, A.J., Goldberger, J.: Training deep neural-networks based on unreliable labels. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2682–2686. IEEE (2016)
3. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 2, pp. 60–65. IEEE (2005)
4. Paranhos da Costa, G.B., Contato, W.A., Nazare, T.S., Batista Neto, J.E.S., Ponti, M.: An empirical study on the effects of different types of noise in image classification tasks. In: XII Workshop de Visão Computacional (WVC 2016) (2016)
5. Dodge, S., Karam, L.: Understanding how image quality affects deep neural networks. In: 2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX). pp. 1–6 (Jun 2016)
6. Ghifary, M., Kleijn, W.B., Zhang, M.: Deep hybrid networks with good out-of-sample object recognition. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5437–5441. IEEE (2014)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)
8. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images (2009)
9. Kylberg, G., Sintorn, I.M.: Evaluation of noise robustness for local binary pattern descriptors in texture classification. EURASIP J. Image and Video Processing 2013, 17 (2013)
10. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
11. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011)
12. Ponti, M., Nazaré, T.S., Thumé, G.S.: Image quantization as a dimensionality reduction procedure in color and texture feature extraction. Neurocomputing 173, 385–396 (2016)
13. Seltzer, M.L., Yu, D., Wang, Y.: An investigation of deep neural networks for noise robust speech recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 7398–7402. IEEE (2013)
14. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2014)
15. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806 (2014)