A Comparative Study About How Image Quality Influences Convolutional Neural Networks

Domonkos Varga[1], Tamás Szirányi[2]

[1] Department of Networked Systems and Services, Budapest University of Technology and Economics, Budapest, Hungary
[2] MTA SZTAKI, Institute for Computer Science and Control, Budapest, Hungary

Abstract

Deep learning in computer vision has been applied to many domains such as image classification, handwritten character classification, pedestrian detection, automatic colorization of grayscale image, content-based image retrieval, etc. While the advantages of deep architectures are widely accepted, the limitations and theoretical background are not satisfactorily researched. In this paper, we provide an evaluation of seven state-of-the-art Convolutional Neural Networks for image classification under different visual distortion types. Namely, we consider nine types of quality distortions: salt & pepper noise, median filtering, average filtering, disk filtering, periodic noise in x- and y-direction, zero-mean Gaussian noise, JPEG compression, and JPEG2000 compression. Our results indentify the distortion types that deteriorate heavily the classification performance. Furthermore, the published results may provide good funds for developing neural networks that are robust to quality distortions.

1. Introduction

Deep learning is part of machine learning algorithms that utilize a cascade of multiple layers of nonlinear processing units for feature extraction and transformation. Each successive layer applies the output from the previous layer as input. Comparing with other "shallow" methods, a deep architecture has more levels of nonlinear operations. Most modern deep architectures are based on an artificial neural network, although they can also consist of latent variables such as Deep Belief Networks[1] or Deep Boltzmann Machines[2].

Deep learning has gained a continously increasing popularity since AlexNet[11] was introduced by Krizhevsky et al. Consequently, deep learning has been sweeping across the research and the industry, as evidenced by the success of different deep architectures in various domains such as computer vision, natural language processing, speech recognition, audio recognition, bioinformatics, etc. In computer vision, deep learning techniques have captured severe attention because they have produced state-of-the-art results in many domains such as image classification[3], handwritten character classification[4], pedestrian detection[5], automatic colorization of grayscale images[6], content-based image retrieval[7], etc.

While the advantages of deep architectures are widely accepted, the limitations and theoretical background are not well researched. In this paper, we introduce an evaluation of seven state-of-the-art deep learning models for image classification under different visual distortions (salt & pepper noise, median filtering, average filtering, disk filtering, periodic noise, zero-mean Gaussian noise, JPEG compression, JPEG2000 compression). The published results may provide good funds for developing neural networks that are robust to quality distortions.

The remaining parts of this paper is organized as follows. We begin with an overview of seven state-of-the-art deep learning models in Section 2. Data processing and experimental setup are described in Section 3. Furthermore, in this section we introduce the results and analysis. Finally, we draw the conclusions in Section 4.

2. Background

In this section we give an overview about the evaluated neural networks. A neural network is a network of simple elements called neurons, which receive input, change their internal state (activation) according to that input, and produce output depending on the input and activation. The network forms by connecting the output of certain neurons to the input of other neurons forming a directed, weighted graph. The weights as well as the functions that compute the activation can be modified by a process called learning which is governed by a learning rule[8].

Starting with LeNet-5[9], CNNs have had a standard architecture. It is composed of one or more convolutional layers with fully connected layers on top. Furthermore, it capitalizes on tied weights and pooling layers. This type of architecture allows CNN to process two- or three-dimensional data such as grayscale and RGB images. Unfortunately, this concept did not take off in the 1980s and 90s because it could not produce a competitive performance due to various reasons such as lack of training data and computing power. In addition, the advent of Support Vector Machines[10] (SVM) for learning tasks, accompanied by solid theoretical foundations and a convex optimization formulation, seemed to be a better solution.

a) AlexNet: Krizhevsky et al.[11] revived the interest in CNNs by winning the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 and by introducing AlexNet. AlexNet was trained on a subset of ImageNet database which consists of 1.2 million 256×256 RGB images belonging to $1,000$ categories. AlexNet has five convolutional layers, three pooling layers, and two fully-connected layers with approximately 60 million free parameters. Furthermore, Krizhevsky et al.[11] introduced the Rectified Linear Unit (ReLU) activation function and they found that ReLU decreases the training time since it is faster than the conventional sigmoid or tanh function. Moreover dropout layers[12] were implemented in order to avoid overfitting. The whole architecture were trained using batch stochastic gradient descent, with specific values for momentum and weight decay.

b) VGG16 and VGG19: The Oxford Visual Geometry Group (VGG) proposed the VGG network in 2014[3]. In contrast to AlexNet's 11×11 filters in the first layer, this model strictly used 3×3 filters with stride and pad of 1, along with 2×2 maxpooling layers with stride 2. The reasoning in the paper[3] was that the combination of two 3×3 convolutional layers has an effective receptive field of one 5×5 convolutional layer. The authors utilized the ReLU activation function and trained using batch gradient descent.

c) GoogleNet: It adopted several ideas from the Network in Network (NIN) concept[13] and is based on the Inception modules. GoogleNet[14] was the first model that deviated from the general approach of simply stacking convolutional and pooling layers on top of each other in a sequential structure. Furthermore, the authors[14] also emphasized that they put special attention to memory and power usage. Namely, stacking of convolutional and fully-connected layers and adding huge numbers of filters has a computational and memory cost, as well as an increased chance of overfitting.

e) Inception V3: As we mentioned, the "Inception" micro-architecture was introduced by Szegedy et al.[14] and the original architecture was called *GoogleNet*. On the other hand, subsequent releases have been called Inception vN where N stands for the version number determined by Google[15].

e) Residual Network (ResNet): It[16] won the 2015 championship on three ImageNet competitions - image classification, object localization and object detection. The main challenge in training deep neural networks is that accuracy deteriorates with the increasing depth of the network. ResNet introduced the so-called residual learning approach in order to overcome the difficulty of training deep networks. The main idea behind a residual block is that an input x goes through a convolution - ReLU - convolution series. Subsequently this result is then added to the original input x. The authors pointed out in their paper[16] that "it is easier to optimize the residual mapping than to optimize the original, unreferenced mapping". Another advantage of the residual block is that during the backward pass of backpropagation[17], the gradient flows easily through the computational graph because there are addition operations which distributes the gradient information in the network.

In order to compare these models, we collected the accuracy values reported in the literature and determined the number of parameters. The results reported in the literature can be seen in Table 1. The output of a network is a probability for each class. These probabilities can be arranged to deliver a vector of predicted classes with decreasing probability. The top-1 accuracy measures the accuracy by comparing the best prediction with the proper class. The top-5 accuracy labels a prediction as correct if the correct class is in the best five predicted classes. The reason that top-5 accuracy is often reported is that for some images in the dataset there are multiple objects in the image.

3. Experimental results

We consider nine types of distortions: salt & pepper noise, median filtering, average filtering, disk filtering, periodic noise in x-direction, periodic noise in y-direction, zero-mean Gaussian noise, JPEG compression, and JPEG2000 compression.

Salt-and-pepper noise is considered, for which a certain amount of the pixels in the image are either black or white. Given the noise density ($0 \leq d \leq 1$) as probability that a pixel is corrupted. In our experiments the noise density was varied from 0.0005 to 0.195. The main idea of *median filter* is to run through the image pixel by pixel, replacing each pixel with the median of neighboring pixels. On the other hand, *average filter* replaces each pixel with the average of neighboring pixels. In our experiments the kernel size was varied from 3×3 to 17×17 in case of median and average filtering. A *disk filter* is a circular averaging filter (pillbox) within the square matrix of size $2 \cdot r + 1$ where r stands for radius. We varied the radius from 1 to 11 in the experiments. An image affected by *periodic noise* looks like a repeating pattern had been added to the original image. In this survey,

Table 1: Comparison of the examined Convolutional Neural Networks.

Model	Top-1 accuracy	Top-5 accuracy	Input size	Number of parameters	Depth
AlexNet[11]	0.625	0.83	$227 \times 227 \times 3$	$60,965,224$	8
VGG16[3]	0.715	0.901	$224 \times 224 \times 3$	$138,344,128$	23
VGG19[3]	0.727	0.910	$224 \times 224 \times 3$	$143,667,240$	26
GoogleNet[14]	0.79	0.93	$224 \times 224 \times 3$	$11,193,984$	21
Inception v3[15]	0.78	0.94	$299 \times 299 \times 3$	$23,851,784$	159
ResNet-50[16]	0.759	0.929	$224 \times 224 \times 3$	$25,636,712$	50
ResNet-101[16]	0.775	0.94	$224 \times 224 \times 3$	$45,765,453$	101

we applied sinusoidial pattern with amplitude A in x- and y-direction. The amplitude was varied from 0.01 to 50. For $JPEG$ compression, the quality parameter was varied from 1% to 99%. A quality value of 100% is equivalent to the original uncompressed image. For $JPEG2000$ compression, the compression ratio was varied from 5 to 500. A compression ratio of 1 represents the original uncompressed image. Figures 1 - 9 show samples of corrupted images.

The test was carried out on a subset of ImageNet 2014 database's validation set. Namely, we randomly chose 20 categories from the available $1,000$ categories. Furthermore, we selected 20 images from the examined categories. For each image we generated additional images with varying levels of quality distortions as described in the previous paragraph.

We consider two types of measure: top-1 accuracy and top-5 accuracy. If the classifier's top guess is the correct answer (e.g., the highest score is for the "cat" class, and the test image is actually of a cat), then the correct answer is said to be in the top-1. If the correct answer is at least among the classifier's top 5 guesses, it is said to be in the top-5. The top-1 accuracy is the percentage of the time that the classifier gave the correct class as the highest score. The top-5 accuracy is the percentage of the time that the classifier included the correct class among its top five guesses.

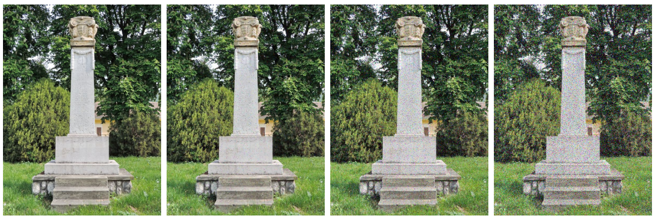
Figure 10 and 11 show the results of our experiment. Table 2 shows top-1 and top-5 acc. measured on the undistorted images. All of the networks are very sensitive to salt & pepper noise, median filtering, average filtering, disk filtering, and Gaussian noise. Even low amount of these noise and distortion types can reduce the classification performance significantly. This decrease is due to the fact that these distortion types removes the texture of the images. Furthermore, CNNs look for texture to classify images. On the other hand, the networks are robust to moderate periodic noise. Surprisingly, all networks are very robust to JPEG and JPEG2000 compression. In the case of JPEG compression, we have to set the quality to 15% in order to produce significant degradation in the classification performance. In the case of JPEG2000 compression, we have to set the compression rate to 400 (very high level of compression) to halve the classification performance. This means that the user can be sure that deep architectures will perform well on JPEG or JPEG2000 compressed images assuming that quality level or compression is not in the extremely low range.

Inception v3[15] appears to be more robust and resilient than the other state-of-the-art networks. One obvious solution to increase the robustness of networks is to put low quality images into the training database. Actually, GoogleNet[14] and Inception v3[15] were trained on images with slight color perturbations to add a plus regularization to the networks. In spite of this, the classification performance of GoogleNet[14] deteriorates at the same rate as VGG16[3] or ResNet50[16]. On the other hand, we experienced stronger robustness by Inception v3[15] than by the other state-of-the-art networks. To sum it up, our results showed that Inception v3[15] had the best classification accuracy and robustness to all types of noises except for JPEG2000 compression.

4. Conclusions

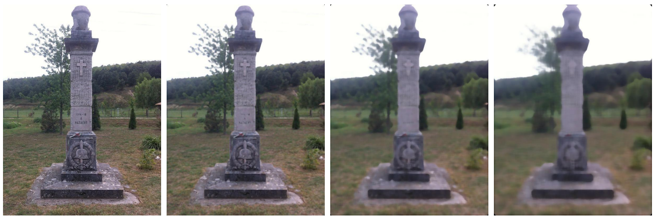
In this paper, we introduced an evaluation of seven state-of-the-art Convolutional Neural Networks for image classification under different visual distortion types. To this end, we took seven state-of-the-art networks and considered nine types of quality distortions such as salt & pepper noise, median filtering, average filtering, disk filtering, periodic noise in x- and y-direction, zero-mean Gaussian noise, JPEG compression, and JPEG2000 compression. Our results showed that Inception v3[15] had the best performance both on undistorted and distorted images. Furthermore, all networks show significant robustness to JPEG and JPEG2000 compression and they are all sensitive to filtering, salt & pepper noise, and Gaussian noise.

In our future work we plan to investigate other important state-of-the-art networks in a similar way such as DenseNet[18] and MobileNet[19]. Furthermore, we want to investigate the possible benefits of training on low quality images.



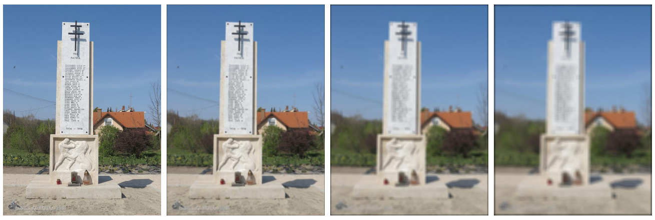
(a) Undistorted. (b) $d = 0.01$. (c) $d = 0.05$. (d) $d = 0.15$.

Figure 1: Salt & pepper noise added to images where d denotes the noise density. This affects $d \times \#Pixels$.



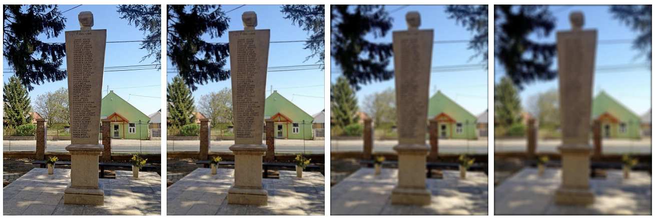
(a) Undistorted. (b) $k = 5$. (c) $k = 11$. (d) $k = 17$.

Figure 2: Median filtering performed on images with $k \times k$ sized kernels.



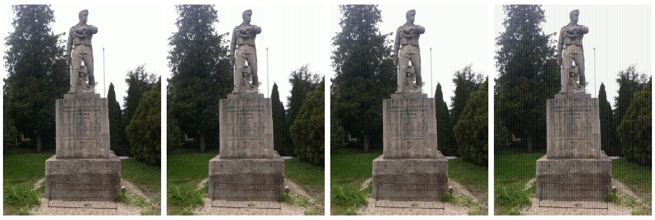
(a) Undistorted. (b) $k = 5$. (c) $k = 11$. (d) $k = 17$.

Figure 3: Average filtering performed on images with $k \times k$ sized kernels.



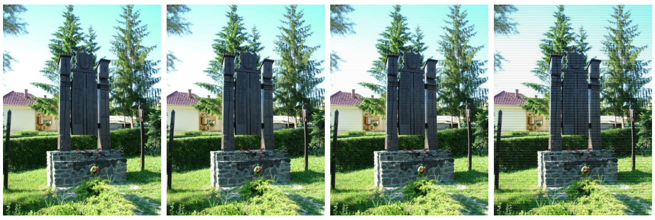
(a) Undistorted. (b) $r = 2$. (c) $r = 6$. (d) $r = 10$.

Figure 4: Disk filtering performed on images with radius r.



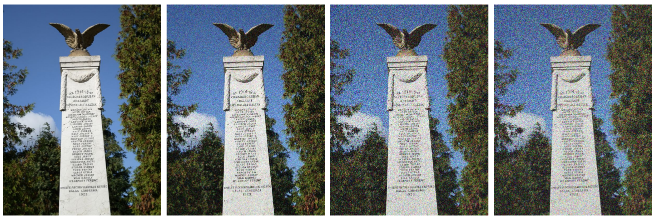
(a) Undistorted. (b) $A = 2$. (c) $A = 10$. (d) $A = 30$.

Figure 5: Periodic noise in x direction with amplitude A.



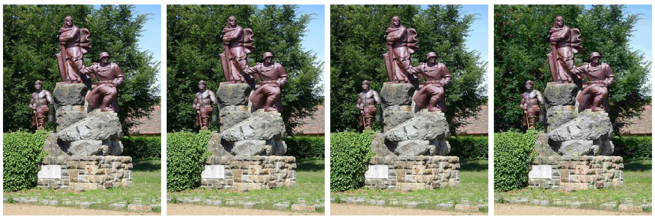
(a) Undistorted. (b) $A = 2$. (c) $A = 10$. (d) $A = 30$.

Figure 6: Periodic noise in y direction with amplitude A.



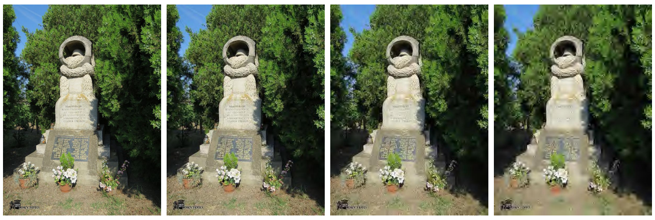
(a) Undistorted. (b) $\sigma = 0.02$. (c) $\sigma = 0.07$. (d) $\sigma = 0.1$.

Figure 7: Zero-mean Gaussian noise.



(a) Undistorted. (b) $q = 90\%$. (c) $q = 40\%$. (d) $q = 10\%$.

Figure 8: JPEG compressed.



(a) Undistorted. (b) $CR = 20$. (c) $CR = 100$. (d) $CR = 330$.

Figure 9: JPEG2000 compressed.

Table 2: Top-1 Accuracy & Top-5 Accuracy measured on the undistorted images.

	AlexNet[11]	VGG16[3]	VGG19[3]	ResNet-50[16]	ResNet-101[16]	GoogleNet[14]	Inception v3[15]
Top-1 Accuracy	0.66	0.745	0.74	0.785	0.815	0.755	0.825
Top-5 Accuracy	0.74	0.85	0.85	0.875	0.87	0.875	0.895

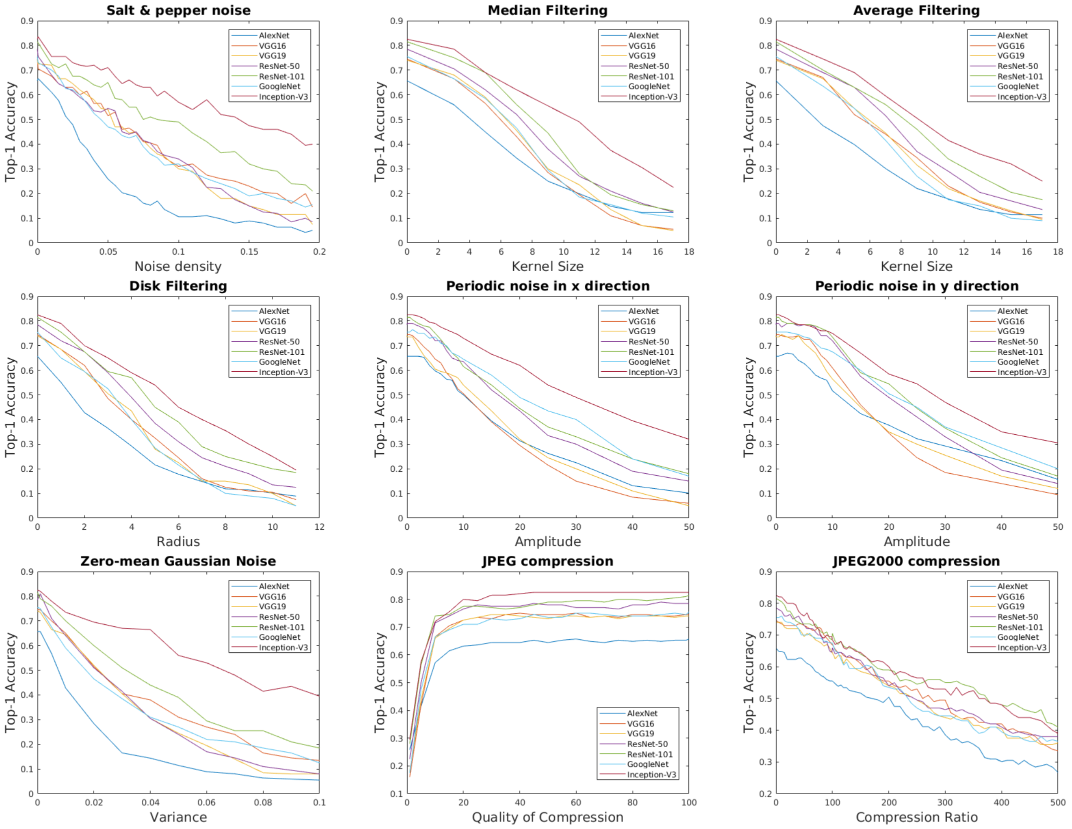


Figure 10: Top-1 accuracy rates under different visual distortions.

References

1. G. Hinton. Deep belief networks. *Scholarpedia*, **4**(5):5947, 2009. 1

2. D. Ackley, G. Hinton, and T. Sejnowski. A learning algorithm for Boltzmann machines. *Readings in Computer Vision*, pp. 522–533, 1987. 1

3. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ArXiv preprint ArXiv:1409.1556*, 2014. 1, 2, 3, 6

4. D. Cireşan and U. Meier. Multi-column deep neural network for offline handwritten chinese character classification. *International Joint Conference on Neural Networks*, pp. 1–6, 2015. 1

5. E. Bochinski, V. Eiselein, and T. Sikora Training a convolutional neural network for multi-class object detection using solely virtual world data. *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 278–285, 2016. 1

6. Y. Xiao, P. Zhou, and Y. Zheng. Interactive Deep Colorization With Simultaneous Global and Local Inputs. *ArXiv preprint ArXiv:1801.09083*, 2018. 1

7. D. Varga and T. Szirányi. Fast content-based image retrieval using convolutional neural network and hash function. *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 002636–002640, 2016. 1

8. A. Zell. *Simulation Neurale Nezte.* Addison-Wesley Bonn, 1994. 2

9. Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel. Backpropaga-

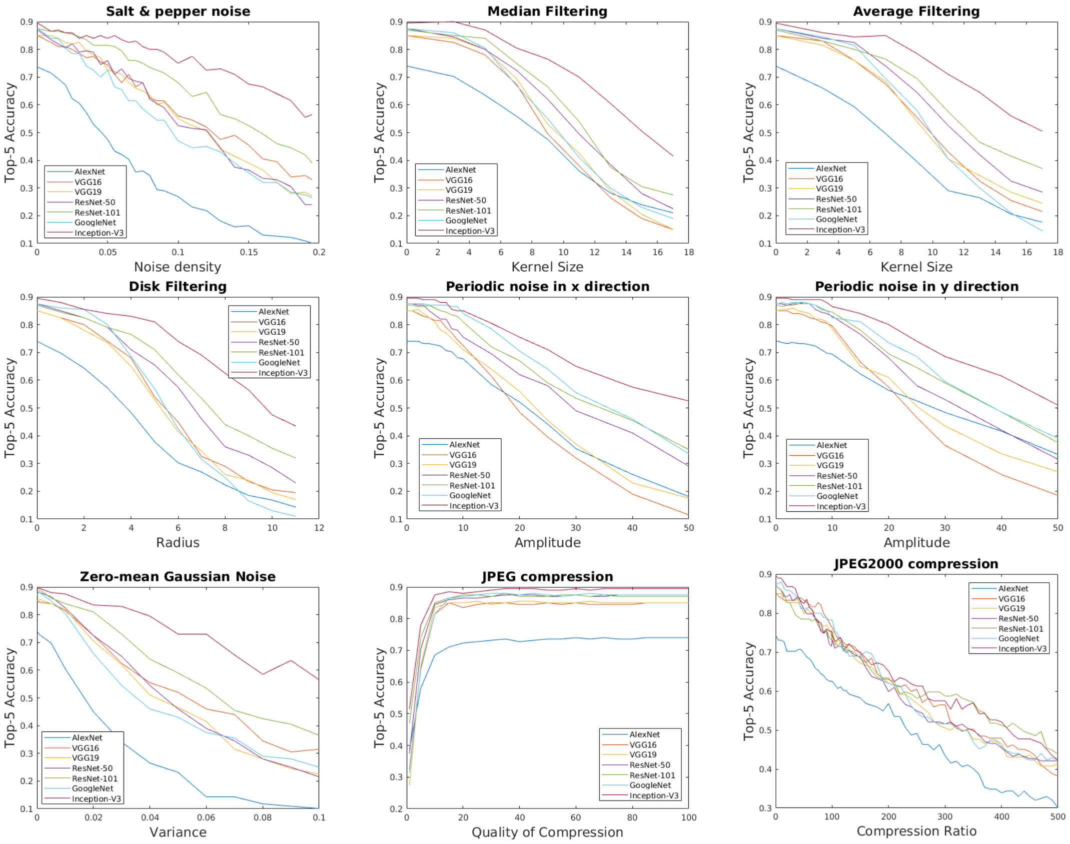


Figure 11: Top-5 accuracy rates under different visual distortions.

tion applied to handwritten zip code recognition. *Neural Computation*, **1**(4):541–551, 1989. 2

10. C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, **20**(3):273–297, 1995. 2

11. A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012. 1, 2, 3, 6

12. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, **15**(1):1929–1958, 2014. 2

13. M. Lin, Q. Chen, and S. Yan. Network in network. *ArXiv preprint ArXiv:1312.4400*, 2013. 2

14. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015. 2, 3, 6

15. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016. 2, 3, 6

16. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. 2, 3, 6

17. R. Hecht-Nielsen. Theory of the backpropagation neural network. *Neural Networks*, **1**(Supplement-1):445–448, 1988. 2

18. G. Huang, Z. Liu, K.Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3, 2017. 3

19. A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and A. Hartwig. MobileNets: Efficient conventional neural networks for mobile vision applications. *ArXiv preprint ArXiv:1704.04861*, 2017. 3