Signal Extraction for Classification of Noisy Images Compressed using Autoencoders

Dorsaf Sebai

Cristal Laboratory
ENSI
Manouba, Tunisia
dorsaf.sebai@ensi-uma.tn

Nour Missaoui

INSAT
Centre Urbain Nord
Tunis, Tunisia

Asma Zouaghi

INSAT
Centre Urbain Nord
Tunis, Tunisia

ABSTRACT

The world is experiencing an increasing boom in computer vision. This is more and more used in many domains such as robotics, medicine, industry, security systems, etc. In this context, Deep Neural Networks (DNNs) have great capabilities and are widely used. Convolutional Neural Networks (CNNs) present a particular class of DNNs that is most commonly leveraged to analyzing visual imagery. However, CNN performances completely depend on two main issues. The first issue is related to the quality of the images generated by capture cameras. All images captured by remote sensors and modern imaging systems are practically noisy, which can prevent the image from being correctly classified and identified by a CNN. The second issue is the throughput available for the transmission of the large amount of data between capture sensors and units processing CNNs. A seamless transmission can be ensured by compression techniques that help reducing the size of data, while affording the required quality for computer vision algorithms. Since lossy compression of noise-free and noisy images differ from each other, this work firstly raises the question of CNNs resilience to noisy images compression using the particular autoencoders. We secondly propose a method that aims to improve this resilience so that CNNs can achieve better classification performances. The compressed noisy images are passed, as a test set, along a model that is learnt from a noise dataset. The subtraction of the so captured noise from the noisy images is then performed to extract the useful signal to classify. This will be first work, where we learn the autoencoder from the noise sample, and not the noisy sample, while denoising. Obtained results prove the efficiency of the proposed method.

Keywords
Autoencoder, noise, classification, noise dataset.

1 INTRODUCTION

Computer vision enhances the understanding of the information depicted by images. However, the capture of the environment through sensors leads to various kinds of voluminous images and videos. Being an unsupervised feature extraction technique, AutoEncoders (AE) can be particularly useful for computer vision algorithms, as they can address the images compression to reduce the transmission delay between sensors and processing units. Further, their feature extraction asset allows not to lose details that cannot be seen by human eye, but are picked up by CNNs. However, autoencoders are generally trained assuming that the input images are artifact-free. This is not the case in real computer vision application scenarios, where digital images

are subject to a wide range of noises. Here, we have to note that autoencoders can be effectively used for noise removal, if the desired output from decoder is fixed to be similar to original images. But, this cannot be exploited if the noise is introduced during image acquisition. In this case, we do not have pristine, i.e., artifact-free, images to use them as desired output of the autoencoder. The output is instead fixed to be similar to noisy images. In this context, the contributions of this paper are as follows :

- Study of the impact of the autoencoder-based noisy images compression on the performance of image classification using CNNs. We investigate in training both vanilla and deep autoencoders. In-depth analyses of involving different noise types of low, medium and high levels in the training process are also provided.

- The inclusion of a signal processing autoencoder in CNNs to get rid of the noise and preserve only the high valued signal elements, which will consequently improve the accuracy of noisy images classification and make the classifier more suitable. The

proposed method operates with no ground truth images available and improves the generalization ability of existing deep denoisers to several real noise models and image types.

The rest of this paper is organized in the following way. Section 2 is dedicated to related work enumeration. In Section 3, we detail the learning architecture used for the image compression and classification. Discussions of test results on pristine and noisy compressed images are also presented. In Section 4, we propose a method that aims to enhance the image classification performance while extracting clearer objects to be detected by CNNs. Evaluation of the proposed method performances along with comparative results are presented in Section 5. Finally, Section 6 concludes the paper.

2 CONTRIBUTIONS

Since classification presents the most common step of almost all computer vision applications, our first aim is to study the impact of noisy image compression using autoencoder on CNN classification performances. This is particularly important as a handful of works, e.g., [1] [2] [3], only study the impact of standard compression techniques, e.g., JPEG and JPEG 2000. However, artifacts of autoencoder-based compression methods have different characteristics from distortions caused by classical codecs [4], especially when images are noisy. Further, existing studies consider noise and compression artifacts separately, whereas we aim to jointly assess the impact of noisy images compression on the CNN performance. This aligns well with the real-world computer vision applications where images are often both noisy and voluminous.

In order to remove the noise harmful impact on image classification, several works have been proposed ranging from common filters, e.g., Gaussian, Median and Bilateral, to more elaborated efforts. The denoising domain is ever evolving, and a recent addition in this regard is the use of deep neural networks, based on CNNs [6] [7], autoencoders [5] [8] or both [10]. Some of the referenced methods are non-blind [8] since they learn on a separate training set and use the trained model to denoise new test samples. This makes them fail when the test images to denoise are not of the same kind as the models learnt with. [5] [6] [7] are however blind methods that overcome the shortcoming of the non-blind ones, insofar as they learn the model adaptively from the signal at hand while denoising. The main disadvantage of both blind and non-blind methods is that, one never knows how good the learned autoencoder will generalize to unseen noises [5] [9]. Most of them are good enough for some noises, such as Gaussian and impulse, but generalize poorly to noisy images with more sophisticated noise. Further, some of the aforesaid methods make use of noisy-clean image pairs [7], whereas pristine images are not always available in real-world computer vision applications.

In this paper, we propose an autoencoder-based method that extracts the signal relevant to the classifier from the noisy compressed images. Actually, we are not really proposing a denoiser where the reconstructed image is recovered from its noisy version. We rather propose a method that looks for the information that is useful to improve the accuracy of the classification and injects it to the CNN. We first make use of an autoencoder to learn the noise features from a noise dataset which includes noise images, and not noisy ones. Second, the compressed noisy images are passed, as a test set, along the model that was obtained from the noise feature extraction step. As the model is trained to distinguish the noise features, it will obviously capture the noise that exists in the compressed noisy images. It is sufficient to subtract this captured noise from the noisy images to extract the useful signal to classify. The key strength of our work is five-fold :

- Unlike some existing methods where noise-free images are necessary for them to function, our method does not require pristine images and is noiseless native images independent. This is more realistic as images captured in real-world computer vision applications are often noisy and no ground truth images are available.

- The proposed work jointly considers the noise and compression artifacts. Bearing in mind real life scenarios, it is usual for an image data to be noisy and voluminous so that it requires noise processing and compression together. To the best of our knowledge, there are no methods that address these two issues together.

- To reduce noise, our method excludes the necessity to train the model for the noisy sample itself, i.e., blind methods, or for a noisy sample other than the one to denoise, i.e., non-blind methods. Our model is learnt from the noise dataset. Thus, the proposed method performance would not be impacted by the volume of the training noisy data, either they are large or not.

- Being independent from noisy images, our method does not fail to efficiently recover different types and modalities of unseen images.

- Most of the existing methods, not to say all of them, perform well only when the noise level and type, present in the training and test noisy images, are same or differ only a little. In our case, this is no longer required. It is enough to prepare a noise dataset that contains the desired noise levels and
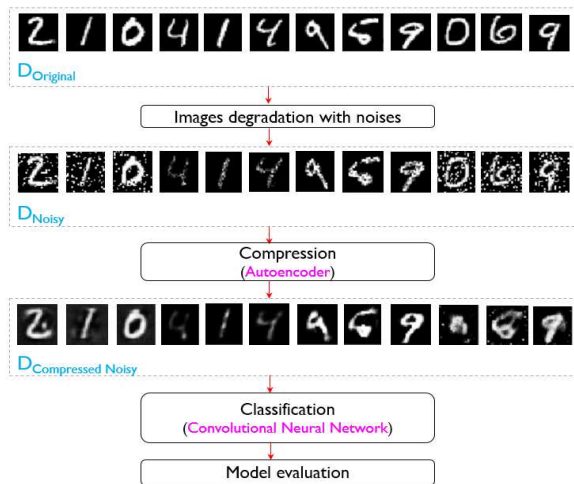
Figure 1: Steps of the learning architecture: MNIST as an example of benchmark dataset.

types. This will be first work, where we learn the autoencoder from the noise sample, and not from the noisy sample, while denoising.

3 CLASSIFICATION OF COMPRESSED NOISY IMAGES

3.1 Learning architecture

Fig. 1 depicts the steps of our learning architecture applied on different benchmark test datasets :

1. We construct the dataset D_{Noisy} that includes the distorted images by applying different common types of noises to original images of the $D_{Original}$ dataset.

2. We construct the $D_{NoisyCompressed}$ dataset that includes the compressed images of the D_{Noisy} dataset using different autoencoder configurations.

3. We pass the images of the $D_{NoisyCompressed}$ dataset along a state-of-the-art CNN in order to classify them.

4. According to results of step 3, we analyze how autoencoders can deal with the injected noises whose undesirable effects, such as artifacts, unrealistic edges, unseen lines, corners, blurred objects and disturbed background scenes, can impact the classification accuracy of CNNs. We remember that the autoencoder here is used for compression and not as a denoiser. Indeed, the noise is introduced during the capturing so that we do not hold pristine images.

The learning architecture shown in Fig. 1 is evaluated on three well-known $D_{Original}$ benchmark datasets, namely Mixed National Institute of Standards and Technology (MNIST), i.e., black and white images,

Fashion MNIST, i.e., grayscale images, and CIFAR-10, i.e., color images. For each of the abovementioned datasets, we construct the D_{Noisy} training data by injecting different types and levels of noise into the pristine training $D_{Original}$ images. We are particularly interested in 5 types of commonly encountered noises, namely the Gaussian as an amplifier noise, Poisson as a shot noise, Uniform as a quantization noise, Speckle as a multiplicative noise and Salt & Pepper as an impulse noise. All these noise types are fairly presented in D_{Noisy} dataset, i.e. each of them presents 20% of the total samples. Furthermore, three levels, corresponding to standard deviation (σ) values of 10, 50, and 100, are considered for each noise. This training on a mixture of samples with multiple types of noises at different levels, rather than a certain type and level, will provide the deep neural networks models, autoencoder and CNN in our case, with stronger generalizing ability.

3.2 Used autoencoders and CNNs

The so obtained D_{Noisy} dataset is the entry of an autoencoder in order to generate the $D_{NoisyCompressed}$ dataset of Fig. 1. For MNIST and Fashion MNIST, 60.000 and 10.000 of the noisy images are used for the training and test, respectively. For CIFAR-10, 50.000 and 10.000 are the number of images used for training and test. Two different settings of the autoencoder are considered. The former presents a vanilla autoencoder with a fully-connected code layer of size 32. The latter presents a deeper autoencoder with 3 encoding fully-connected layers of respective sizes equal to 128, 64 and 32. We note that a fully-connected layers based autoencoder effectively operates for MNIST and Fashion MNIST, but not for CIFAR-10. Thus, we opt for a CNN-AE for this latter. Unlike simple handwritten digits and clothing items, there are many more features and details to extract from each CIFAR-10 image. To ensure the abovementioned settings, we make use of CNN-AEs with 1 and 3 encoding convolutional layers. These two settings are trained using Adam optimizer with a batch size 128 for 100 epochs. All our experiments use the initial learning rate of 0.1 which decays for every 20 epochs with an exponential rate of 0.1. The Mean Squared error (MSE) is used as loss function. Here, we remind that the reconstructed images are aimed to be as close as possible to noisy ones since as already mentioned no noise-free images are available.

The $D_{NoisyCompressed}$ dataset, output of the aforesaid autoencoder, is then provided as the input data to learn the CNN model. We use 70% of the $D_{NoisyCompressed}$ samples for each CNN training epoch. The remaining 30% are exploited for the test. To make the future comparisons fair, we look for the CNN model that fits each benchmark dataset. If the same CNN would be used for all datasets, this would produce unbiased results as the CNN architecture is optimized for one dataset and not

Table 1: Training strategies.

Strategies	Training images	Noise levels	AE depth
1	Pristine	-	1
2	Pristine	-	3
3	Noisy (Gaussian, Poisson, Uniform, Speckle and S&P)	10	1
4	Noisy (Gaussian, Poisson, Uniform, Speckle and S&P)	10	3
5	Noisy (Gaussian, Poisson, Uniform, Speckle and S&P)	50	1
6	Noisy (Gaussian, Poisson, Uniform, Speckle and S&P)	50	3
7	Noisy (Gaussian, Poisson, Uniform, Speckle and S&P)	100	1
8	Noisy (Gaussian, Poisson, Uniform, Speckle and S&P)	100	3

for the other. Thus, we opt for three CNN models that are optimized for each of our datasets MNIST, Fashion MNIST, and CIFAR-10 and allow respective accuracies of 99.8%, 93%, and 90%. These models are largely detailed in [11], [12] and [13].

3.3 Results and discussion

Eight different training strategies are implemented to evaluate the impact of the noisy compressed images on one of the main computer vision applications : classification. All these strategies share the same architecture of Fig. 1, but present as depicted in Table 1 different combinations of training images, noise levels and autoencoder depths. For each strategy, the classification performances will be then analyzed when the CNNs in [11], [12] and [13] are trained on pristine and compressed noisy images of MNIST, Fashion MNIST, and CIFAR-10 datasets, respectively. As indicated in section 3.2, the compression is carried out using the two autoencoder settings, namely depth 1 and 3. As a shorthand, we refer to these settings as AE_D1 and AE_D3. Classification performances are evaluated in Table 2 in terms of accuracy (%), and will serve as reference values in section 5 to assess our proposed noise recovery method. We note that each cell in Table 2 is composed of two lines that respectively correspond to the autoencoder settings : the value in the first line is related to AE_D1, whereas the value in the second line corresponds to AE_D3. The numbers after the down arrow correspond to the difference between accuracy values obtained for noisy compressed images and pristine compressed ones.

Results of Table 2 exhibit that training strategies with AE_D1, i.e., 1, 3, 5 and 7 of Table 1, allow better classification performances than AE_D3, i.e., 2, 4, 6 and 8, on both pristine and noisy images. Furthermore, the classification performances of the CNN model, trained on noisy compressed images, decrease when the noise level increases. Although expected, we could not measure how much this decay would be if we had not carried out these experiments. Compared to pristine images, the accuracy of CNN for compressed noisy

MNIST images decreases, at σ value of 10, by 9.27% for AE_D1 and 7.90% for AE_D3. The decrease is about 11% and 10.96% for σ value of 50, and 14% and 14.41% for σ of 100. For Fashion MNIST, the accuracy of the CNN for compressed noisy images decreases, when compared to pristine images at level 10, by 9.60% and 8.68% for the respective AE_D1 and AE_D3. Even with the σ value 50 where the noise is still moderate, the accuracy drops almost 16.01% for AE_D1 and 17.29% for AE_D3. The same observation also stands for the CIFAR-10 dataset.

4 EFFICIENT CLASSIFICATION SCHEME FOR AE-BASED COMPRESSED NOISY IMAGES

Following the study of the noisy image autoencoder-based compression impact on classification, the proposed method to enhance obtained results is presented in this section. Compared to the flowchart of Fig. 1, we include a new step as a part of the enhanced learning architecture. As shown in Fig. 2a, this step consists in extracting the signal from the mixture of image and noise. It is actually decomposed to three sections, namely noise features learning, noise extraction then signal subtraction (*cf.* Fig. 2b) :

- **Noise features learning :** First, we construct a noise dataset that includes noise, and not noisy, images. Typically, they are images that contain noise without any other content, e.g., objects, contours, and texture. The constructed dataset is consisted of 45000 images of the same size than the training dataset images. All the five noises as well as their three levels, specified in section 3.1, are equally presented in the dataset. Second, we train a vanilla autoencoder, as defined in section 3.2, on our noise dataset to generate a model that captures the noise features.

- **Noise extraction :** The $D_{NoisyCompressed}$ dataset is no more passed, as it is, along the CNN for classification. A noise extraction step is added in order to subtract the noise from the images before their classification. We typically consider the $D_{NoisyCompressed}$ as a test set for the model that was obtained from the noise features learning step. Since the model is trained on a noise dataset, its parameters, namely weights and biases, are adjusted in a way that they represent noise features. Therefore, passing $D_{NoisyCompressed}$ dataset along the noise model as a test set would generate images that are similar to those of noise ones.

- **Signal subtraction :** The output images of the noise extraction step are subtracted from the $D_{NoisyCompressed}$ images in order to extract the signal that is useful for the classification, i.e., the absolute

	Pristine images	Compressed noisy images		
		$\sigma = 10$	$\sigma = 50$	$\sigma = 100$
MNIST	99.57	90.30 ↓ 9.27	88.57 ↓ 11.00	85.57 ↓ 14.00
	96.43	88.53 ↓ 7.90	85.47 ↓ 10.96	82.02 ↓ 14.41
Fashion MNIST	93.21	83.61 ↓ 9.60	77.20 ↓ 16.01	70.18 ↓ 23.03
	91.51	82.83 ↓ 8.68	74.22 ↓ 17.29	68.03 ↓ 23.48
CIFAR-10	90.04	79.10 ↓ 10.94	73.26 ↓ 16.78	69.16 ↓ 20.88
	86.69	75.28 ↓ 11.41	70.05 ↓ 16.64	67.72 ↓ 18.97

Table 2: Classification accuracy (%) of pristine and compressed noisy images for **MNIST**, **Fashion MNIST**, and **CIFAR-10** datasets. Each cell is composed of two lines that are respectively related to **AE_D1** and **AE_D3** settings.

value of the subtraction. Here, we note that our aim is not the image denoising. We rather aim to take a neat signal from the image, and subtract the noise that can disrupt the CNN classification results. As we can observe in the last rectangle of Fig. 2b, the images backgrounds are also removed while carrying out the subtraction. This aligns with what we have just affirmed, the fact that the proposed method is not aimed for denoising. It is for object shape preservation to enhance classification results of CNNs. The image background removal will not affect the CNN performances since what is important here is the object to be classified.

5 EXPERIMENTATIONS

5.1 Comparison with candidate methods

Performances of the proposed method are shown through accuracy in Table 3 for classification of compressed noisy images of MNIST, Fashion MNIST, and CIFAR-10 datasets. Each cell of Table 3 is composed of seven lines. The first line corresponds to classification performances without noise processing (*cf.* Fig. 1) that were obtained from experiments of section 3.3. The six remaining lines are respectively related to classification performances with noise processing using the proposed method (*cf.* Fig. 2a), the 5×5 median, gaussian and bilateral filters, and Agostinelli et al. method [8]. For our method, the AE_D1 setting has been considered as autoencoder depth, as it considerably performs better than the AE_D3 one according to results of Table 2. Like our method, Agostinelli et al. [8] method is based on autoencoders. As a non-blind approach, we train the model in Agostinelli et al. [8] method on two separate training data that are respectively different from and more or less of similar content to test MNIST, Fashion MNIST and CIFAR-10 samples.

We typically use ImageNet [14] and EMNIST [15] whose images content respectively resembles the CIFAR-10 dataset and MNIST variants. However,
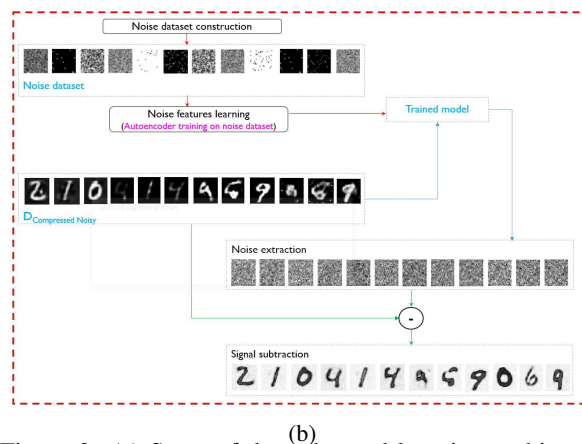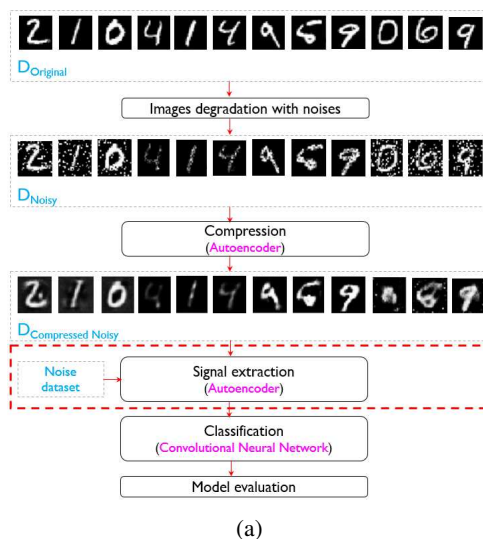

(a)

(b)

Figure 2: (a) Steps of the enhanced learning architecture : MNIST as an example of benchmark dataset, (b) Flowchart of the proposed signal extraction.

SARS-CoV-2 CT-scan dataset [16] is leveraged as totally different kind of images from our benchmark MNIST, Fashion MNIST, and CIFAR-10 test samples. Being one of the biggest COVID-19 datasets, SARS-CoV-2 CT-scan includes 2482 Computerized Tomography (CT) scans from 120 patients, with 1252 CT scans of 60 patients infected by COVID-19, and 1230 CT scan images of 60 non-infected patients by COVID-19, but presenting other pulmonary diseases. Improvements of the candidate methods over the classification without noise processing are indicated by the values after the up arrows. We remind that the CNNs in [11], [12] and [13] are used for the classification of MNIST, Fashion MNIST and CIFAR-10 for all the candidate methods.

Obtained results of Table 3 show that our method is more robust than its candidates. It succeeds in classifying images with more confident level than filters and Agostinelli et al. [8] method for all the evaluation noise rates and benchmark datasets. For MNIST dataset as example, the proposed method enables the CNN to reach an accuracy of 96.74%, 93.68% and 90.83% for respective levels 10, 50 and 100, meaning an enhancement of 6.44%, 5.11% and 5.26% in comparison with CNN used without any noise processing. These enhancements are higher than those achieved by the gaussian, median and bilateral filters, the most used noise filters. Trained on EMNIST, Agostinelli et al. [8] method (*Agostinelli et al. [8]_EMNIST*) achieves lower classification accuracies than ours. Accuracies are further lower when Agostinelli et al. [8] method is trained on CT (*Agostinelli et al. [8]_CT*) that is not similar to the MNIST test set. In fact, Agostinelli et al. method [8] performances depend on the images it was trained on. They are considerably decreased when the test images to denoise, i.e., MNIST, Fashion MNIST and CIFAR-10, are not of the same kind as the model learnt with, i.e., CT.

5.2 Results per noise type

Until now, our evaluation considered all types of noise together gathered in the test set. As already mentioned, the noise types are fairly presented in D_{Noisy} dataset, where each of them presents 20% of the total samples. In this present section, we assess the CNN performances of Fig. 2a while being trained on the mixture of the noise types, but tested on each type of noise separately. Hence, we can study how damaging a specific type of noise is when compressed by an autoencoder. The obtained results are presented in Fig. 3 in terms of accuracy for MNIST, Fashion MNIST, and CIFAR-10 at AE_D1. As can be observed, Gaussian and Uniform are noises that mostly affect the classification performances as they present the lowest accuracies for all the test datasets and noise levels. In contrary, the Speckle noise leads to the best results.
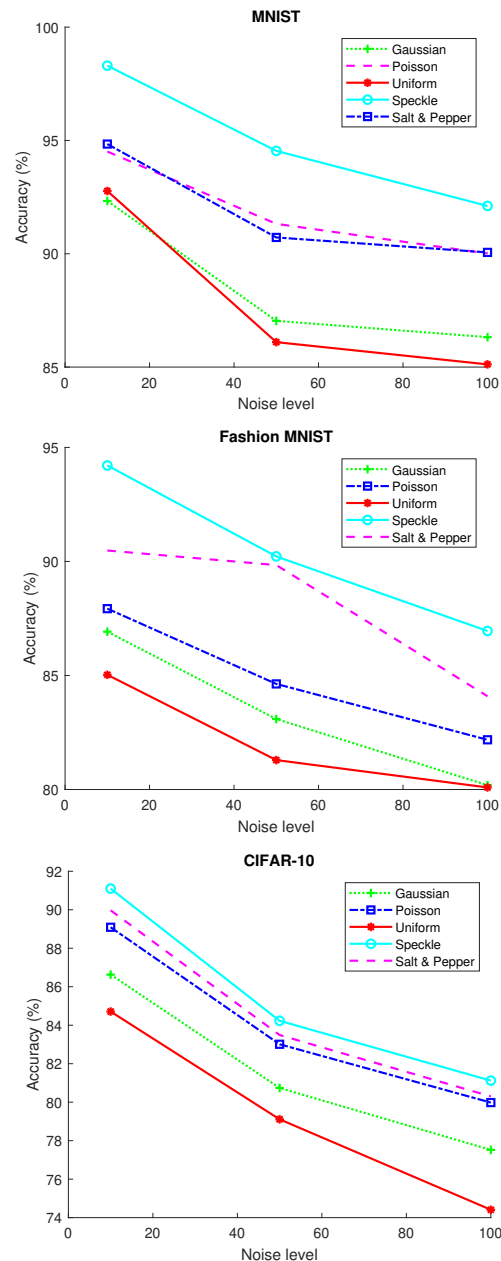

Figure 3: Classification accuracy (%) of compressed noisy images for MNIST *(top)*, (b) Fashion MNIST *(middle)*, and (c) CIFAR-10 *(bottom)* per separate noise type.

6 CONCLUSION

Image classification is a non trivial visual task, especially when it faces the presence of real life inevitable noise and compression artifacts. To address these issues, several researches have been conducted utilizing denoisers to restore original images, from noisy ones, before using CNNs for classification. We rather aim to extract, from noisy images, the pertinent signal useful for classification. The noisy compressed images are passed, as a test set, along a model that is trained on a noise dataset. The so generated images are then sub-

		Compressed noisy images		
		σ = 10	σ = 50	σ = 100
MNIST	w/o Noise processing	90.30	88.57	85.57
	Proposed method	96.74 ↑ 6.44	93.68 ↑ 5.11	90.83 ↑ 5.26
	Gaussian	94.57 ↑ 4.27	92.91 ↑ 4.34	90.25 ↑ 4.68
	Median	93.30 ↑ 3.00	91.28 ↑ 2.71	88.57 ↑ 3.00
	Bilateral	93.53 ↑ 3.23	90.72 ↑ 2.15	88.63 ↑ 3.06
	Agostinelli et al. [8]_EMNIST	94.10 ↑ 3.80	91.31 ↑ 2.74	90.02 ↑ 4.45
	Agostinelli et al. [8]_CT	91.11 ↑ 0.81	89.05 ↑ 0.48	86.41 ↑ 0.84
Fashion MNIST	w/o Noise processing	83.61	77.20	70.18
	Proposed method	90.33 ↑ 6.72	88.21 ↑ 11.01	85.37 ↑ 15.19
	Gaussian	89.18 ↑ 5.57	86.92 ↓ 9.72	84.63 ↑ 14.45
	Median	87.23 ↑ 3.62	85.78 ↓ 8.58	84.11 ↑ 13.93
	Bilateral	88.01 ↑ 4.40	85.45 ↓ 8.25	82.29 ↑ 12.11
	Agostinelli et al. [8]_EMNIST	89.30 ↑ 5.69	86.43 ↑ 9.23	84.14 ↑ 13.96
	Agostinelli et al. [8]_CT	84.44 ↑ 0.83	80.04 ↑ 2.84	77.27 ↑ 7.09
CIFAR-10	w/o Noise processing	79.10	73.26	69.16
	Proposed method	87.29 ↑ 8.19	82.89 ↑ 9.63	79.10 ↑ 9.94
	Gaussian	85.81 ↑ 6.71	81.62 ↑ 8.36	77.06 ↑ 7.90
	Median	84.79 ↑ 5.69	80.10 ↑ 6.84	76.83 ↑ 7.67
	Bilateral	83.55 ↑ 4.45	79.66 ↑ 6.40	75.26 ↑ 6.10
	Agostinelli et al. [8]_ImageNet	85.73 ↑ 6.63	79.55 ↑ 6.29	75.60 ↑ 6.44
	Agostinelli et al. [8]_CT	81.81 ↑ 2.71	75.46 ↑ 2.20	72.11 ↑ 2.95

Table 3: Classification accuracy (%) of compressed noisy images for **MNIST**, **Fashion MNIST**, and **CIFAR-10** datasets. Each cell is composed of seven lines that are respectively related to classification : without (w/o) noise processing, and with noise processing using the proposed method, gaussian filter, median filter, bilateral filter, and Agostinelli et al. [8] trained on datasets respectively similar to, i.e., EMNIST and ImageNet, and different from, i.e., CT, test samples.

tracted from the input noisy compressed images, in order to extract clearer versions that are likely to be correctly classified. Extensive experimental results on different benchmark datasets clearly demonstrate the superior accuracy of our method over main state-of-the-art methods. These promising results make the proposed method potentially useful for computer vision systems, where images are voluminous and highly exposed to several noises. The per noise experiments have been also conducted. It is concluded that the classification performances are susceptible to all noise types, albeit to varying degrees. Gaussian and Uniform, which are amplifier and quantization noises, affect the resilience of CNN when classifying compressed noisy images more than Poisson, Speckle, and Salt & Pepper ones, that are shot, multiplicative and impulse noises. Several future research directions are opened from this study. First, the simple autoencoder investigated in this study is found efficient. Various autoencoders rather than the simple counterparts can also be employed to

check whether the noise model training process improves or not. Second, future researches can be conducted by leveraging the super-resolution techniques in order to retrieve details from noisy compressed images, so that we get better CNN performance.

7 REFERENCES

[1] M. Dejean-Servières, K. Desnos, K. Abdelouahab, W. Hamidouche, L. Morin, and M. Pelcat, "Study of the Impact of Standard Image Compression Techniques on Performance of Image Classification with a Convolutional Neural," Network. HAL Archives, 2018.

[2] Z. Chen, W. Lin, S. Wang, L. Xu, and L. Li, "Image Quality Assessment Guided Deep Neural Networks Training," arXiv:1708.03880v1, 2017.

[3] O. K. Al-Shaykh, and R. M. Mersereau, "Lossy compression of noisy images," IEEE Transactions on Image Processing, vol. 7, issue 12, pp. 1641–1652, 1998.

[4] G. Valenzise, A. Purica, V. Hulusic, and M. Cagnazzo, "Quality Assessment of Deep-Learning-Based Image Compression," IEEE 20th International Workshop on Multimedia Signal Processing, 2018.

[5] A. Majumdar, "Blind Denoising Autoencoder," IEEE Transactions on Neural Networks and Learning Systems, vol. 30, issue 1, pp. 312–317, 2019.

[6] D. Sil, A. Dutta, and A. Chandra, "Convolutional Neural Networks for Noise Classification and Denoising of Images," IEEE Region 10 Conference (TENCON), 2019.

[7] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward Convolutional Blind Denoising of Real Photographs," IEEE Conference on CVPR, 2019.

[8] F. Agostinelli, M. R. Anderson, and H. Lee, "Adaptive multi-column deep neural networks with application to robust image denoising," Advances in Neural Information Processing Systems, 2013.

[9] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising," IEEE Transactions on Image Processing, vol. 26, issue 7, pp. 3142–3155, 2017.

[10] S. S. Roy, Sk. I. Hossain, M. A. H. Akhand, and K. Murase, "A Robust System for Noisy Image Classification Combining Denoising Autoencoder and Convolutional Neural Network," International Journal of Advanced Computer Science and Applications, vol. 9, issue 1, pp. 224-235, 2018.

[11] kaggle.com/elcaiseri/mnist-simple-cnn-keras-accuracy-0-99-top-1. Last accessed : 09/10/2020.

[12] kaggle.com/fuzzywizard/fashion-mnist-cnn-keras-accuracy-93. Last accessed : 09/10/2020.

[13] appliedmachinelearning.blog/2018/03/24/achieving-90-accuracy-in-object-recognition-task-on-cifar-10-dataset-with-keras-convolutional-neural-networks. Last accessed : 09/10/2020.

[14] image-net.org. Last accessed : 19/12/2020.

[15] nist.gov/itl/products-and-services/emnist-dataset. Last accessed : 19/12/2020.

[16] E. Soares, P. Angelov, S. Biaso, M. H. Froes, D. K. Abe, "SARS-CoV-2 CT-scan dataset: A large dataset of real patients ct scans for sars-cov-2 identification," medRxiv, May 2020.