FEW-CLASS LEARNING FOR IMAGE-CLASSIFICATION-AWARE DENOISING

Kei Mamiya, Takamichi Miyata

Chiba Institute of Technology, Japan

ABSTRACT

The accuracy of image classification networks decreases substantially in noisy situations such as low light environments. A simple solution to this problem is to integrate an image denoising network as a preprocessing operation before classification and then to retrain the classifier to improve the classification accuracy. However, this straightforward approach suffers from excessive training time because the denoising network requires the entire dataset to conduct end-to-end training. In this paper, instead of using classification accuracy as a loss function, we propose using the difference of the outputs of the hidden layer of the image classification network. This loss function, known as feature loss or perceptual loss, allows us to train the denoising network using only limited images containing extremely few classes from the dataset. The experimental results show that the proposed method dramatically improves the classification accuracy, when we use only a few classes (from 2.5% to 10% of the original dataset) for training. This approach is effective on previously unseen classes even when the image classifier network has been changed by fine-tuning.

Index Terms— image denoising, image classification, deep learning, fine-tuning

1. INTRODUCTION

A DNN (deep neural network) is a versatile framework for performing a variety of image processing and computer vision tasks. In particular, DNNs have achieved significant successes in image classification [1, 2, 3]. However, a well-known drawback of these classifiers is that they are highly sensitive to low-level image degradations such as noise, blur, and compression artifacts. For example, noise is inevitable when applying a pretrained image classification network to images captured by surveillance cameras operated in low light conditions. However, retraining the image classifier network itself for each type of degradation is prohibitively expensive due to the extremely high computational complexity.

A simple and straightforward solution to this problem is to use a DNN-based image denoising network [4, 5, 6, 7, 8] as a preprocessing step before image classification. A similar approach involves building cascading networks to address low-level vision tasks such as denoising and high-level vision tasks (image classification, image retrieval, semantic segmentation) was proposed in [9, 10, 11].

Liu et al. proposed a cascaded network architecture in [10] that included an image denoising network and an image classification network. Because the training of the denoising network was performed using an end-to-end approach, they adopted cross entropy as their loss function, which requires images representing all the classes in the dataset for training. In the case of ILSVRC 2012, the dataset contains 1,000 classes. Thus, the method proposed in [10] still suffers from long training times. Consequently, a natural question arises: does training the denoising network truly require the entire image dataset?

In this study, we propose a new method for training cascading image denoising and image classification networks using images belonging to only a few classes. Unlike the method in [10], which uses the classification results as the loss function, our proposed method uses the output vector of the hidden layer (feature vector) of the image classification network for the loss function. We train our denoising network to output a feature vector similar to the output feature vector of the original image.

We reveal that only a few classes (from 2.5% to 10%) in the original dataset are sufficient to train a denoising network that can improve the classification accuracy even under severe noise. We also show that the proposed method performs well even when the classification network is fine-tuned to a dataset with completely unknown classes.

The adopted loss function is not new. Using a feature vector for the loss function is called feature loss or perceptual loss [12] and has been used in many studies [11, 13]. In particular, the study in [11] uses feature loss to train a cascaded network similar to ours. However, in these methods, feature loss is used to reconstruct visually pleasing outputs. To the best of our knowledge, this is the first study to reveal that feature loss enables us to train a classification-aware denoising network from very few classes. The contributions of this paper are summarized as follows:

- Using feature loss allows the training process to improve the classification accuracy of the denoising network using only images that represent extremely small percentages of the classes in the dataset.

- Even if the image classification network has been changed by fine-tuning on images belonging to totally unseen classes, our denoising network can improve the classification accuracy on noisy images.

2. PREVIOUS WORK

In [10], Liu et al. proposed connecting a denoising network and an image classification network and revealed that cascading two networks is mutually beneficial: improving both the classification accuracy and the visual appearance of the denoised image. They used VGG16 as the image classification network and a U-Net [14]-based network as the denoising network. The loss function of the cascaded network, which is the weighted sum of cross entropy and mean squared error (MSE), is defined as follows:

$$L(\Theta) = \sum_i e(C(D(\boldsymbol{y}_i;\Theta)), \boldsymbol{u_i}) + \lambda \|D(\boldsymbol{y}_i;\Theta) - \boldsymbol{x}_i\|_2^2, \quad (1)$$

where $\boldsymbol{x_i}$ is a i-th clean image, and $\boldsymbol{y_i}$ is a corresponding noisy image obtained by adding Gaussian noise with a standard deviation of σ to each pixel in $\boldsymbol{x_i}$. Let $\boldsymbol{u_i}$ be a one-hot vector of the ground truth label of $\boldsymbol{x_i}$. VGG16 and the image denoising network are denoted by $C(\cdot)$ and $D(\cdot;\Theta)$, respectively, and Θ represents all the trainable parameters of D. The first term of the loss function is the cross entropy $e(\cdot)$ between the classification result of the noisy image and the corresponding grand truth $\boldsymbol{u}$. Thus, we can improve the classification accuracy by minimizing this term. The second term is used to improve the subjective quality of the denoised images. The balance between these terms is controlled by the parameter λ.

In this paper, unlike the study in [10], we focus on improving the classification accuracy in noisy environments. Setting $\lambda = 0$ in Eq. (1) gives

$$L_{\mathrm{c}}(\Theta) = \sum_i e(C(D(\boldsymbol{y}_i;\Theta)), \boldsymbol{u_i}), \quad (2)$$

and is appropriate for our objective. We refer to the denoising network trained with the loss function shown in Eq. (2) as the *classification-oriented denoising network* (CDnN) and use CDnN as the comparison method in our experiments, which are described in Section 4. As mentioned earlier, because CDnN uses cross entropy as the loss function, it requires datasets with complete class sets when training the denoised network.

3. PROPOSED METHOD

In this section, we introduce a new learning method for the denoising network to improve the classification accuracy of the cascaded network that requires only a few data classes. Our cascaded network, shown in Fig. 1, consists of the denoising network and the image classification network. To overcome the problem of CDnN, we use the output of the hidden layer in VGG16 (feature vector) to train the denoising network. We use a 4,096-dimensional vector immediately before the 1,000-class classification of VGG16 as our feature vector. We split VGG16 into two parts before and after the final layer and denote them as *featureNet* and *classNet*. Thus, our feature vector can be considered as the output of featureNet. The loss function of the proposed method is defined as follows:

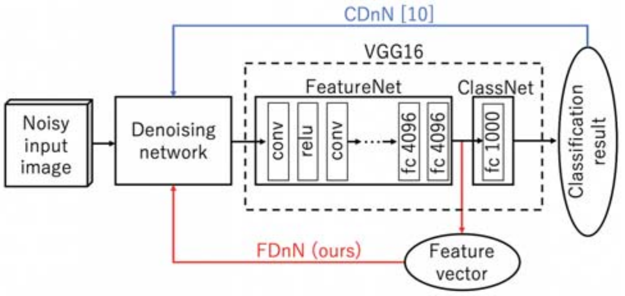


Fig. 1. The network architecture of the proposed method. We use a feature vector for our loss function instead of the final classification result.

$$L_{\mathrm{f}}(\Theta) = \sum_i \|F(D(\boldsymbol{y}_i;\Theta)) - F(\boldsymbol{x}_i)\|_2^2, \quad (3)$$

where $F(\cdot)$ is a map corresponding to featureNet. Eq. (3) shows that our loss function is defined by the MSE between the denoised and the original image. We denote the denoising network that is optimized by the above loss function as the *feature-oriented denoising network* (FDnN).

Because the feature vector potentially contains more information than the classification results, we expect that training with the loss function shown in Eq. (3) should improve the classification accuracy for unknown classes. We also conduct some preliminary experiments to find the best layer from which to extract the feature vector; however, we found no other layers that improve the classification performance other than the 4,096-dimensional feature vector described above.

4. EXPERIMENTAL RESULTS

We performed two experiments to evaluate the effectiveness of our proposed method. The first experiment used a VGG16 model pretrained on the ILSVRC2012 training set. This experiment showed that our proposed method is effective even when images containing only a few classes are used to train the denoising network. The second experiment showed that the denoising network trained by the proposed method works well even with a classification network fine-tuned with totally unknown classes. This second experiment also used a VGG16 model fine-tuned on the Food-101 dataset [15].

	FeatureNet	ClassNet	Denoising network
Sec 4.1	$\mathcal{I}_{1000}$	$\mathcal{I}_{1000}$	$\mathcal{I}_{25}, \mathcal{I}_{50}, \mathcal{I}_{75}, \mathcal{I}_{100}$
Sec 4.2	$\mathcal{I}_{1000}$	$\mathcal{F}_{101}$	$\mathcal{I}_{25}, \mathcal{I}_{50}, \mathcal{I}_{75}, \mathcal{I}_{100}$

Table 1. A summary of the datasets and subset used to train the networks. The dataset symbols are described in Sec. 4. Throughout the paper, we use the same fixed parameters for featureNet, which was provided by the official PyTorch website. Note that for the experimental setting in Sec. 4.2, the denoising network was totally agnostic for the Food-101 dataset.

Denoising network		W/o denoising	DnCNN [4]	CDnN (special case of [10])				FDnN (ours)			
Dataset for training		-	-	$\mathcal{I}_{25}$	$\mathcal{I}_{50}$	$\mathcal{I}_{75}$	$\mathcal{I}_{100}$	$\mathcal{I}_{25}$	$\mathcal{I}_{50}$	$\mathcal{I}_{75}$	$\mathcal{I}_{100}$
$\sigma = 0$	top-1	71.59	-	-	-	-	-	-	-	-	-
	top-5	90.39	-	-	-	-	-	-	-	-	-
$\sigma = 15$	top-1	57.83	65.99	64.92	66.51	66.97	67.39	**68.35**	**68.63**	**68.51**	**68.90**
	top-5	81.03	86.90	86.16	87.38	87.68	87.83	**88.36**	**88.47**	**88.36**	**88.64**
$\sigma = 30$	top-1	33.72	56.17	56.51	59.96	60.45	61.29	**63.20**	**63.89**	**63.96**	**64.28**
	top-5	58.51	80.06	79.91	82.50	83.09	83.52	**84.91**	**85.35**	**85.45**	**85.60**
$\sigma = 45$	top-1	13.92	47.80	47.88	52.58	53.66	54.29	**56.67**	**57.80**	**58.60**	**58.87**
	top-5	30.63	72.80	72.84	76.83	77.85	78.32	**79.94**	**80.89**	**81.41**	**81.63**
$\sigma = 60$	top-1	4.70	38.54	40.27	45.08	46.21	48.03	**49.34**	**50.87**	**51.91**	**52.36**
	top-5	12.93	63.63	65.37	70.19	71.41	73.24	**73.93**	**74.81**	**76.31**	**76.64**

Table 2. Performance comparison with the top-1 and top-5 classification accuracy of VGG16 (trained on ILSVRC2012) under noise. The denoising network with our training method (FDnN) shows the best performance in every combination of the number of classes used for training and noise levels. The best results for a given noise level and number of classes are shown in bold.

Throughout this paper, we denote the ILSVRC2012 and Food-101 datasets as $\mathcal{I}$ and $\mathcal{F}$, respectively, and denote a subset made by randomly extracting n classes from dataset $\mathcal{A} \in \{\mathcal{I}, \mathcal{F}\}$ as $\mathcal{A}_n$. Table 1 summarizes the datasets and subsets used to train the networks in each experiment.

We used the *denoising convolutional neural network* (DnCNN) [4], which is known for its high performance, as an image denoising network. Because the DnCNN estimates the noise component from a noisy image, a corresponding denoised image can be obtained by subtracting the estimated noise from the noisy image. We trained parameters of DnCNN for color image denoising task by using official implementation and the dataset mentioned in [4] for each noise level and used that as the initial parameters of our denoising network.

4.1. Image denoising as the preprocessing of VGG16

We used the ILSVRC2012 training and validation sets to train the cascaded network and the test images to evaluate the classification accuracy. A preprocessing operation (cropping and pixel value shifting) for VGG16 [1] was applied during the training procedure.

First, we obtained noisy input images by adding additive Gaussian noise (σ = 15, 30, 45, 60) to all the images in the dataset. The top-1 and top-5 classification accuracies of VGG16 for the noisy images are shown in the leftmost column of Table 2. The results show that the accuracy decreases rapidly as the noise level increases.

Then, we applied DnCNN [4], CDnN, and FDnN (ours) to the noisy images and obtained the corresponding classification accuracies on the denoised images. We use sets of classes (25, 50, 75, and 100) randomly chosen from the ILSVRC2012 training set. Then, we chose 650 and 10 images randomly from each class for training and validation, respectively. The learning rate for the denoising network was initial set to 0.001 and multiplied by 0.2 after 30, 60 and 90 epochs. The training was completed at 100 epochs. We chose the model that yielded the best validation accuracy for subsequent testing. We trained all the denoising networks with each noise level.

Table 2 shows all the resulting classification accuracies, revealing the effectiveness of the proposed method (VGG16 + FDnN) on every class and at each noise level. Notably, when only a few classes are used for training, the accuracy gap between FDnN and CDnN increases.

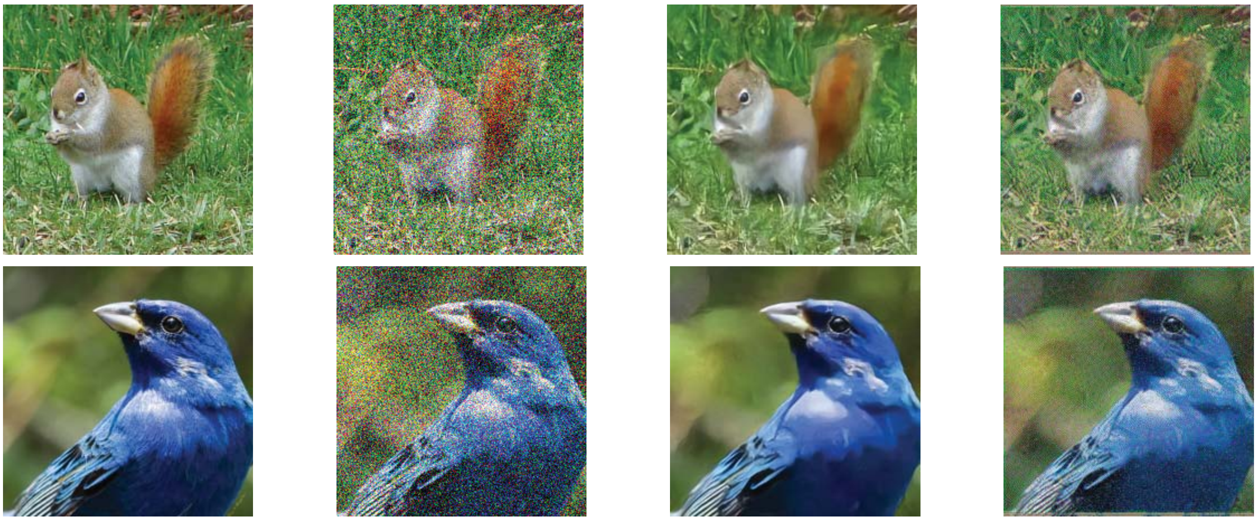
Fig. 2 shows the original, noisy, and two denoised images by DnCNN and FDnN. DnCNN (c), successfully denoises the images, but they appear somewhat over-smoothed. In contrast, the denoised images by FDnN (d), appear to still contain some noise. Table 2 shows that the accuracy is higher when using FDnN than when using DnCNN. These results suggest that the remaining noise contributes to improving the classification accuracy.

4.2. Image denoising as a preprocessing step for a fine-tuned classification network

We also evaluated the performance of our proposed method with a fine-tuned network. In this experiment, we used Food-101 [15] for the target fine-tuning dataset. This dataset con-

Denoising network	W/o denoising	DnCNN [4]	CDnN (special case of [10])				FDnN (ours)			
Dataset for training			$\mathcal{I}_{25}$	$\mathcal{I}_{50}$	$\mathcal{I}_{75}$	$\mathcal{I}_{100}$	$\mathcal{I}_{25}$	$\mathcal{I}_{50}$	$\mathcal{I}_{75}$	$\mathcal{I}_{100}$
$\sigma = 0$	60.00	-	-	-	-	-	-	-	-	-
$\sigma = 15$	36.70	50.40	50.92	53.00	52.83	54.26	**56.04**	**56.14**	**56.32**	**56.90**
$\sigma = 30$	18.27	38.10	41.33	44.25	45.07	47.07	**49.59**	**50.43**	**49.90**	**50.74**
$\sigma = 45$	8.58	29.75	34.37	37.37	38.32	38.32	**43.04**	**43.28**	**44.74**	**44.48**
$\sigma = 60$	4.13	22.81	29.59	31.34	32.31	34.68	**36.16**	**36.76**	**38.73**	**38.16**

Table 3. Comparison of the top-1 classification accuracy for the fine-tuning experiment on the Food-101 dataset. Note that CDnN and FDnN are agnostic for Food-101. The results clearly show that FDnN always outperforms DnCNN and CDnN. The best results under a given noise level and same number of classes are shown in bold.



(a) Clean image

(b) Noisy image ($\sigma = 60$)

(c) Denoised image by DnCNN [4]

(d) Denoised image by FDnN (ours)

Fig. 2. Denoised images from the ILSVRC2012 validation dataset. The FDnN was trained by $\mathcal{I}_{100}$.

sists of 101 classes of foods; each class is represented by 1,000 images. We split these images and used 750 as training and 250 as testing data.

To fine-tune the VGG16 network, we fixed the parameters of featureNet (see Fig. 1) and only retrained the classNet portion to adapt it to the Food-101 dataset. We denote the fine-tuned VGG16 as Food CNN. For this experiment, we evaluate top-1 accuracies. The remainder of the experiment was performed using the same procedure described in the previous subsection.

Table 3 shows that our Food CNN can correctly classify clean images with 60% accuracy. However, the accuracy decreases as the noise level increases. Again, we applied DnCNN, CDnN, and FDnN to the noisy images before performing classifications with Food CNN (the parameters of CDnN and FDnN are identical to those described in the previous subsection). Table 3 clearly shows that our FDnN achieves the best performance compared with DnCNN and CDnN despite that the FDnN is agnostic for Food-101.

5. CONCLUSION

In this paper, we proposed a new method for training an image-classification-aware denoising method that requires only a very small subset of the classes in the original dataset. The network architecture is a cascaded arrangement of an image classification network (VGG16 in this study) and the image denoising network (DnCNN). We trained this network using a loss function based on a feature vector extracted from VGG16.

The experimental results show that the proposed method drastically improves the classification accuracy under noisy conditions even when we apply the denoised network to a dataset containing completely unseen classes. Our method provides a plug-and-play denoising network for image classification networks in various severe environments without retraining the whole classification network. In this paper, we selected the classes used to train DnCNN randomly from the original dataset; however, in future work we will attempt to discover an efficient method for selecting these classes.

6. REFERENCES

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp. 770–778.

[3] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 7132–7141.

[4] 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, no. 7, pp. 3142–3155, July 2017.

[5] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.

[6] T. Remez, O. Litany, R. Giryes, and A. M. Bronstein, "Class-aware fully convolutional gaussian and poisson denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 11, pp. 5707–5722, Nov. 2018.

[7] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 2472–2481.

[8] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: A persistent memory network for image restoration," in *Proceedings of International Conference on Computer Vision*, 2017.

[9] B. V. Somasundaran, R. Soundararajan, and S. Biswas, "Image denoising for image retrieval by cascading a deep quality assessment network," in *IEEE International Conference on Image Processing*, Oct. 2018, pp. 525–529.

[10] D. Liu, B. Wen, X. Liu, Z. Wang, and T. Huang, "When image denoising meets high-level vision tasks: A deep learning approach," in *International Joint Conference on Artificial Intelligence, IJCAI-18*, July 2018, pp. 842–848.

[11] D. Liu, B. Wen, J. Jiao, X. Liu, Z. Wang, and T. S. Huang, "Connecting image denoising and high-level vision tasks via deep learning," *arXiv:1809.01826*, 2018.

[12] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*, 2016.

[13] S. Prasad Mudunuri, S. Sanyal, and S. Biswas, "Genlrnet: Deep framework for very low resolution face and object recognition with generalization to unseen categories," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018.

[14] O. Ronneberger, P .Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*. 2015, vol. 9351 of *LNCS*, pp. 234–241, Springer.

[15] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 – mining discriminative components with random forests," in *European Conference on Computer Vision*, 2014.