2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

Improving Low-Resolution Image Classification by Super-Resolution with Enhancing High-Frequency Content

Liguo Zhou
Department of Informatics
Technical University of Munich
Garching, Germany
liguo.zhou@tum.de

Guang Chen
School of Automative Studies
Tongji University
Shanghai, China
guangchen@tongji.edu.cn

Mingyue Feng
Department of Informatics
Technical University of Munich
Garching, Germany
mingyue.feng@tum.de

Alois Knoll
Department of Informatics
Technical University of Munich
Garching, Germany
knoll@in.tum.de

***Abstract*—With the prosperous development of Convolutional Neural Networks, currently they can perform excellently on visual understanding tasks when the input images are high quality and common quality images. However, large degradation in performance always occur when the input images are low quality images. In this paper, we propose a new super-resolution method in order to improve the classification performance for low-resolution images. In an image, the regions in which pixel values vary dramatically contain more abundant high frequency contents compared to other parts. Based on this fact, we design a weight map and integrate it with a super-resolution CNN training framework. During the process of training, this weight map can find out positions of the high frequency pixels in ground truth high-resolution images. After that, the pixel-level loss function takes effect only at these found positions to minimize the difference between reconstructed high-resolution images and ground truth high-resolution images. Compared with other state-of-the-art super-resolution methods, the experiment results show that our method can recover more high frequency contents in high-resolution image reconstructing, and better improve the classification accuracy after low-resolution image preprocessing.**

I. Introduction

Nowadays, the Convolutional Neural Networks (CNNs) have dramatically facilitated the progress of high-level visual understanding tasks [1], [2], [3] and low-level image processing tasks [4], [5], [6]. However, in real world applications, CNNs always fail to deal with visual understanding tasks in harsh conditions, such as low resolution, low luminance and adverse weathers. The most direct solution to resolve this problem is preprocessing the low quality images captured in adverse conditions. This connects the low-level image processing methods and the high-level visual understanding methods. Vidal *et al*. [7] and Yang *et al*. [8] have completed a lot of trials in this direction. Their work reveals that the current preprocessing methods can improve the performance of visual algorithms on some of the low quality images, but also degrade their performance on some other images. For instance, image super-resolution (SR), which can scale up a low-resolution (LR) image to a high-resolution (HR) image, is supposed to increase the accuracy of visual classification, but sometimes they can also introduce artifacts and affect the judgement of classifier. In general, the existing preprocessing methods can only slightly improve the performance of classification or even degrade the performance.

In this paper, we focus on improving the performance of visual classification for low-resolution images by super-resolution method. The research on single image super-resolution has lasted a long term, and many typical methods have been proposed, such as interpolation-based method [9], reconstruction-based method [10] and example-based method [11]. In recent years, the CNN-based super-resolution methods [12], [13], [14] outperform the above methods and achieve a very high peak signal-to-noise ratio (PSNR) in validation. Most of the state-of-the-art CNN-based SR methods share the same training process and the same loss functions. Generally speaking, the training process usually contains two main steps: 1. generating the training sample pairs. 2. learning a mapping to reconstruct a HR image from a LR image. A training sample pair consists of a ground truth HR image and a LR image, and the LR is downscaled from the HR. In the training process, the network takes the LR as input and outputs a reconstructed HR image. Then the loss function calculates the difference between the reconstructed HR and the ground truth HR. Based on this computed result, the network updates its parameters by back propagation. In order to achieve a higher PSNR, the loss functions usually equally take all pixels into consideration, which can result in blur and artifacts. However, in image processing, it is well known that a high PSNR cannot always represent the high subjective quality. We think pixels located at different positions should be treated differently, just like humans naturally focus on textures and edges when recognizing some objects.

In an image, textures and edges are the parts in which pixel values vary dramatically. In the frequency domain, they correspond to the high frequency contents which help us to distinguish different objects. We suppose that the performance of a CNN classifier can be improved if more high frequency contents are recovered in HR image reconstructing. To this end, we propose a modified image super-resolution method which focus on high frequency contents reconstruction. Firstly, the pixels which contain more high frequency contents are selected out from ground truth HR image. Then, we give them a high weight during the training loss calculating process. To determine that which pixels contain more abundant high

978-1-7281-8808-9/20/$31.00 ©2020 IEEE

2020 25th International Conference on Pattern Recognition (ICPR) | 978-1-7281-8808-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPR48806.2021.9412876

frequency contents, we propose a method to extract a weight map from a ground truth HR image. This weight map has the same shape of the ground truth HR image. In this ground truth HR image, if a pixel's value is much more different from its surrounding pixels, the weight map will have a larger value at its corresponding position. In a follow-up process, on the weight map, the positions which are occupied by large enough values will be denoted by the number '1', and all other positions will be denoted by the number '0'. As a result, each number '1' in the weight map corresponds to a high frequency pixel in the ground truth HR image. After that, we train the network according to this weight map. The network focuses on minimizing the difference between reconstructed HR and ground truth HR at positions of the high frequency pixels, and it ignores the differences at positions of the low frequency pixels. This means we only use a part of the training data in network training, but our method can obtain more high frequency contents and the same or even better PSNR compared to those methods which use 100% training data. Furthermore, for those classification tasks with low-resolution input images, after scaling up the LR images by our SR method and retraining the network, the network can achieve a higher classification accuracy.

Experiment Source Code can be found in https://github.com/zhouliguo/Low-Resolution-Image-Classification

II. Related Work

A. Image Processing joint Visual Understanding

With the widely use of computer vision techniques, low-level image processing tasks and high-level visual understanding tasks are inevitably to be combined in some comprehensive applications. Zhou *et al*. [15] and Liu *et al*. [16] propose frameworks that can use results of high-level vision tasks to improve performance of the low-level image processing.

In [15], visual classification results is used to supervise the training of image super-resolution network. The top layers of classification CNNs are supposed to extract perceptual information of objects. In this method, a well reconstructed HR image should not only have a high PSNR with respect to the ground truth HR image, but also be the same as outputs of top layers in classification CNNs. The authors connect a SR network with a classification network, and optimize the SR network by minimizing the difference between the reconstructed HR image and the ground truth HR image, and the difference of the classification network's outputs when inputting reconstructed HR image and ground truth HR image synchronously. [16] proposes a joint network for image denoising. This network is similar to the network in [15] , both of them can gain higher PSNR and more visual satisfaction.

B. Robust Visual Understanding in Wild

Many outdoor camera platforms, like UAVs, surveillance cameras and outdoor robots, are of advantage to the society and people in general. However, the images captured by them are always unclear and cannot be interpreted automatically. To facilitate the research in this area, Vidal *et al*. [7] proposed a benchmark dataset, namely the UG^2, which contains images collected from three difficult real-world scenarios: uncontrolled videos taken by UAVs and manned gliders, as well as controlled videos taken on the ground. This benchmark aims at validating whether or not image restoration and enhancement techniques improve visual classification and object detection performances.

Similar to [7], Yang *et al*. [8] collected three benchmark datasets in real-world poor visibility environments, such as bad weathers (haze, rain) and low light. These datasets focus on object detection in the haze, face detection in the low light condition and zero-shot object detection with raindrop occlusions, and aim to evoke a comprehensive discussion and exploration about whether and how the low-level image processing techniques can benefit the high-level automatic visual recognition tasks in various scenarios.

III. Method

A. Image Super-Resolution Training Framework

The mainstream deep learning based image super-resolution methods are learning a mapping model to reconstruct a high-resolution image from a low-resolution image. The whole training framework is showed in Figure 1 (solid lines connected part). During the training process, the Super-Resolution CNN takes the LR image X as its input, and outputs a reconstructed HR image Y'. Then the difference between Y' and the ground truth HR image Y is calculated as training loss which can be used for guiding the back propagation to optimize the weight parameters of the network. The most commonly used loss functions in image super-resolution methods are $L1$ or $L2$ distance [17]. In this paper, we use the $L1$ distance, and the loss functions are defined as:

$$l = \frac{1}{whc}\sum_{i=1}^{w}\sum_{j=1}^{h}\sum_{k=1}^{c}\left|Y\left[i,j,k\right] - Y^{'}\left[i,j,k\right]\right| \tag{1}$$

$$l_b = \frac{1}{N}\sum_{n=1}^{N} l^{(n)} \tag{2}$$

where l denotes the loss of one pair of Y' and Y. w, h and c denote the width, height and channel number of Y and Y' respectively. N and $l^{(n)}$ denote the total number of sample-tuples and the n-th tuple's loss in one training batch respectively, and l_b denotes the average loss of one batch of training sample-tuples.

In conventional deep learning based image super-resolution methods, all image pixels are treated equally in the loss function. In fact, for network optimization, the pixels in the high frequency regions of image play a much more important role than those which exists in low frequency regions. Hence, giving priority to optimize the high frequency regions could make the method to be more efficient. To achieve this, as show in Figure 1, we design a weight map W and add it in the training framework. The weight map has the same shape of Y, it can help the network screen out high frequency regions from

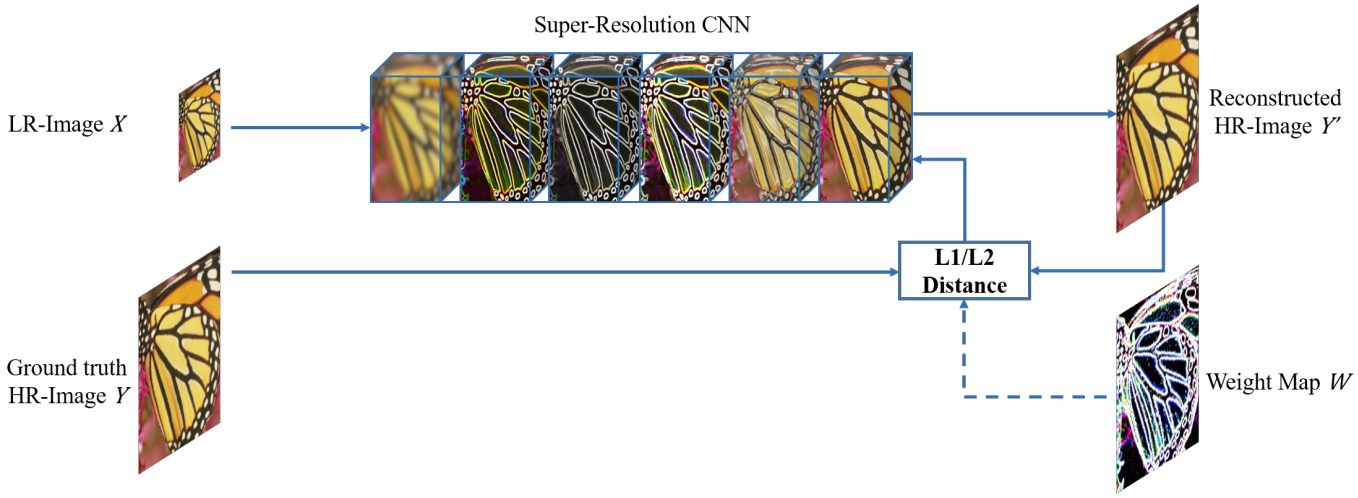


Fig. 1. Our CNN training framework for image super-resolution. We add a weight map in loss function for recovering more high frequency content.

the ground truth HR images, therefore enables the network to focus on optimizing loss at these regions and enhances its capability to obtain more high frequency content. More details about weight map generation are provided in next section. After integrated with weight map, the loss function of the new network changes from (1) to (3):

$$l = \frac{1}{m}\sum_{i=1}^{w}\sum_{j=1}^{h}\sum_{k=1}^{c}\left|(Y[i,j,k] - Y'[i,j,k])W[i,j,k]\right| \quad (3)$$

where m denotes the total number of none-zero values in weight map W.

B. Weight Map Initialization

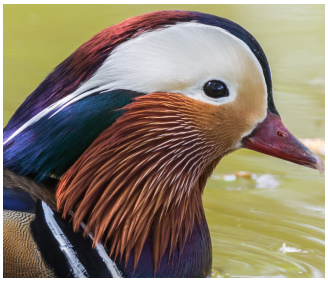
In order to select pixels which contain more high frequency content, we propose a method to assign a weight to per pixel position. Figure 2(b) represents a 3×3 pixel neighborhood in ground truth HR image. Obviously, in this pixel neighborhood, the center pixel p_c has four nearest adjacent pixels which locate on the up, down, left and right side of it respectively. For each pixel neighborhood, equations (4)-(6) are defined for calculating the initial weight value w for p_c:

$$D = \begin{bmatrix} p_c - p_u \\ p_c - p_d \\ p_c - p_l \\ p_c - p_r \end{bmatrix} \quad (4)$$

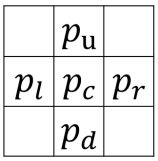
$$D_a = \begin{bmatrix} |p_c - p_u| \\ |p_c - p_d| \\ |p_c - p_l| \\ |p_c - p_r| \end{bmatrix} \quad (5)$$

$$w = D[argmax(D_a)] \quad (6)$$

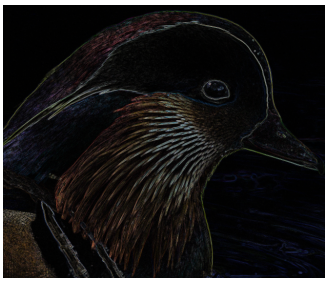
Firstly, the differences between the center pixel p_c and its four nearest adjacent pixels are computed by equation (4). Then the absolute values of these differences are calculated by equation (5). After that, in equation (6), the maximum absolute value is found, and its corresponding difference is selected as the initial weight value w for p_c. For those pixels locating at corners and edges, we calculate their initial weight values with their nearest two or three adjacent pixels if they really exist. Since an image usually has three channels, all channels are processed one by one in the same way.



(a) Ground Truth Image



(b) A Pixel Neighborhood



(c) Initial Weight Map

(d) Modified Weight Map

Fig. 2. (a) represents the ground truth image (cropped from the '0068.png' in training dataset of DIV2K [18]). (b) denotes a pixel neighborhood in ground truth image. (c) is the initial weight map generated by (4)-(6) (the negative values are absolutized for visualization). (d) is the weight map modified by (7) from (c) for pixels selection.

Figure 2(c) shows a sample result of initial weight map calculating. Apparently, the high frequency regions in the original image have an obvious representation in this generated weight map. After weight map initialization, the next step is

to select pixels for training according to this weight map.

C. Weight Map Modification for Training Pixels Selection

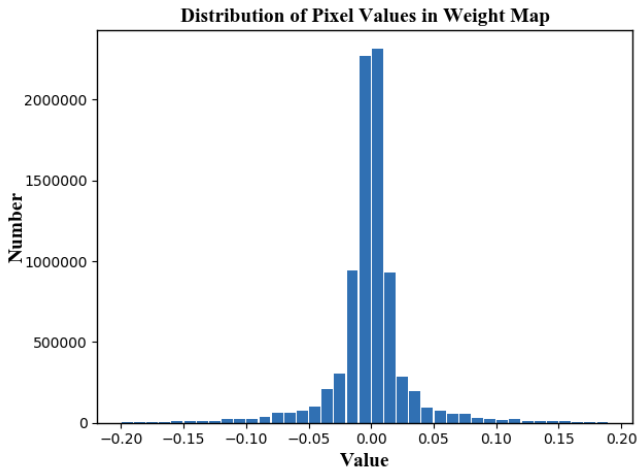


Fig. 3. The distribution of pixel values in initial weight map (normalized to [-0.5,0.5]). The horizontal axis denotes weight value in weight map and the vertical axis denotes weight number. (Generate from the '0068.png' in training dataset of DIV2K [18].)

With the statistic analysis, the distribution of the weight values in a initial weight map approximately obeys the Gaussian distribution. As show in Figure 3, most of the weight values concentrated near the zero point, and the positions of these weight values represent positions of low frequency contents in original image. On the contrary, a few of them have very large values and these weight values are corresponding to the high frequency contents.

The most needed part of ground truth HR image for network training is the high frequency part, so it is necessary to screen out pixels containing high frequency information from the ground truth HR image. For this purpose, we modify the initial weight map (the result of the previous subsection) according to the weight values' Gaussian distribution.

Firstly, the initial weight map values are normalized to a range of [0,1], next the mean μ and the standard deviation σ of their distribution are calculated. Then the weight values are modified by (7):

$$w^{'} = \begin{cases} 0, & w \in [\mu - \alpha\sigma, \mu + \alpha\sigma] \\ 1, & otherwise \end{cases} \tag{7}$$

The parameter α is used for controlling and limiting the total number of zero values in a weight map. A modified weight map looks like the sample in Figure 2(d) and the bright parts of it represent the high frequency content in ground truth image. Finally, this modified weight map is applied to train the network. As a result, in the loss function, only the loss of high frequency pixels are multiplied by the weight value 1, this implies that they have been selected. On the other hand, the loss of low frequency pixels are multiplied by 0 and ignored. Since only the high frequency part of the ground truth HR image contributes to the back propagation, the network can reconstruct more high frequency contents.

IV. Experiment

A. Datasets

In this section, we evaluate the performance of our image super-resolution method and its effects on the subsequent object recognition task on DIV2K [18] and CIFAR [19] datasets respectively. DIV2K dataset consists of 1000 2K resolution RGB images which contain a large diversity of contents. These images are divided into three parts: 800 images for training, 100 images for validation, and 100 images for testing. CIFAR dataset contains 2 subsets: CIFAR-10 and CIFAR-100. The CIFAR-10 subset is composed of 60000 32×32 color images in 10 classes, namely 6000 images for each class. From another point of view, there are 50000 training images and 10000 testing images in CIFAR-10. The CIFAR-100 subset is similar to CIFAR-10, except it has 600 images for each of the 100 classes. In addition, there are 500 training images and 100 testing images in each class.

TABLE I
Performance Comparison of Super-Resolution Methods on DIV2K Dataset

	Proportion of Training Data	PSNR (dB)	SSIM
Bicubic [9]	—	29.91	0.8680
SRCNN [4]	100%	32.97	0.9196
WDSR [14]	100%	34.54	**0.9368**
$\alpha = \frac{1}{7}$	71.73%	**34.55**	**0.9368**
$\alpha = \frac{1}{6}$	67.66%	**34.55**	0.9367
$\alpha = \frac{1}{5}$	**63.07%**	**34.55**	**0.9368**
$\alpha = \frac{1}{4}$	56.87%	34.54	0.9367
$\alpha = \frac{1}{3}$	48.32%	34.52	0.9365

B. Super-Resolution Network Training

Since WDSR [14] is a state-of-the-art CNN-based image super-resolution method, we adopt the network of WDSR-A and integrate it with our weight map in the training process. The number of Residual Blocks [20] in WDSR-A is set to 8. The parameter α is set to 1/3, 1/4, 1/5, 1/6 and 1/7 respectively, and the network is trained on bicubic downscaling ×2 DIV2K images with Tensorflow [21]. We use the same optimizer [22] and hyperparameter setting in [14]. The trained models can scale up the width and height of an image 2 times. Then we measure the PSNR of our model on the validation images. The result is depicted in Table I.

As shown in Table I, different α values represent different amount of non-zero values in the weight map. If α is set to a non-zero value, it means that only a part of the training data is used for model training. On the contrary, if α is set to 0, 100% of the training data are used for training, in this case, the training process is the same as WDSR training. Furthermore, the results in Table I demonstrate that our method can achieve

the same (even better) PSNR and the same SSIM [23] as the original WDSR method in super-resolution task, even though only about half amount of the training data are used for training.

C. Effects on Classification Task

Firstly, the training and test images in CIFAR dataset are upscaled by the models trained in section 4.2. Then we train the classification network on this upscaled dataset. The classification network is a ResNet [20] with 164 layers. The details of this network are presented in Table II. The vectors in the last column represent kernel size, feature channel number and stride size of the layers. For example, the vector '3×3, 16, 1' means the kernel size of this layer is 3×3, it outputs a feature map with 16 channels, and the stride is 1.

To make a comparison, we firstly train the classification network by the original 32×32 images, then train it by the 64×64 images upscaled by bicubic, SRCNN, WDSR and our method (with different α) respectively. Momentum optimizer [24] is used with $momentum = 0.9$. The batch size is set to 64. The learning rate is initialized to 0.1 and is multiplied by 0.1 at 80K, 120K and 160K iterations respectively. The training ends at 180K iterations. For data augmentation, we pad 4 zero-value pixels to the CIFAR-10 images' edges and 8 zero-value pixels to CIFAR-100 for randomly cropping, as well as flip the images horizontally and randomly. All the training images are used in training process. Image whitening is applied to every image in training and testing. The classification results are presented in Table III.

TABLE II
STRUCTURE OF THE CLASSIFICATION NETWORK

Layer Name	Feature Size		Layer Details
Input	32×32	64×64	
Conv 0	32×32	64×64	3×3, 16, 1
Conv 1_x	32×32	64×64	[1×1, 64, 1; 3×3, 64, 1; 1×1, 64, 1] ×18
Conv 2_x	16×16	32×32	[1×1, 128, 2; 3×3, 128, 1; 1×1, 128, 1] ×1 [1×1, 128, 1; 3×3, 128, 1; 1×1, 128, 1] ×17
Conv 3_x	8×8	16×16	[1×1, 256, 2; 3×3, 256, 1; 1×1, 256, 1] ×1 [1×1, 256, 1; 3×3, 256, 1; 1×1, 256, 1] ×17
Average pool	4×4	8×8	2×2, –, 2
10-d/100-d fc, softmax			

The results in Table III show that the bicubic interpolation method cannot significantly improve the performance of classification, in some cases it even reduces the classification accuracy. Meanwhile, the SRCNN and WDSR slightly improve the classification performance on both datasets. On the other hand, our method (with different α) can better improve the classification accuracy than the original WDSR method. More specifically, when α is set to 1/5, the classification accuracy on dataset CIFAR-10 reaches the peak as 95.23%. In addition, when α is set to 1/6, the classification accuracy on dataset CIFAR-100 achieves its highest value as 78.24%.

TABLE III
CLASSIFICATION RESULTS ON CIFAR

	Accuracy	
	CIFAR-10	CIFAR-100
Original Size	94.69%	77.21%
Bicubic (×2)	94.43%	77.59%
SRCNN (×2)	94.78%	77.23%
WDSR (×2)	94.80%	78.12%
$\alpha = \frac{1}{7}$ (×2)	95.15%	78.05%
$\alpha = \frac{1}{6}$ (×2)	95.18%	**78.24%**
$\alpha = \frac{1}{5}$ (×2)	**95.23%**	78.17%
$\alpha = \frac{1}{4}$ (×2)	95.04%	78.06%
$\alpha = \frac{1}{3}$ (×2)	95.05%	78.13%

V. DISCUSSION

Does our method really get more high frequency contents in super-resolution? In this section, we will discuss this question.

A. Comparison of File Size after Differential Coding Compression

Differential coding [25] is a typical method for image compression. It compresses images by utilizing the differences between neighboring pixels. In an image, if the pixel values vary dramatically in most of the pixel neighborhoods, this means that this image contains abundant high frequency contents, and its compressed version will have a large file size. Otherwise, the compressed version will be small in size.

In Table IV, we list file sizes of compressed version of upscaled datasets. These upscaled datasets consist of the reconstructed HR images generated by different super-resolution methods. To compare the amount of high frequency contents these datasets contain, they are compressed into a lossless format, namely PNG (Portable Network Graphic) [26], by differential coding compression. In experiment, these compressed images are generated by OpenCV-Python with default parameters.

As shown in Table IV, the compressed datasets generated from results of the bicubic method always have the smallest size. They are followed by the compressed versions of results of WDSR method, which have the second smallest size all the time. Generally speaking, compared with bicubic and WDSR

TABLE IV
DIFFERENTIAL CODING COMPRESSION SIZE (MB) OF THE UPSCALED DATASETS

	DIV2K	CIFAR-10		CIFAR-100	
	Validation Set	Training Set	Test Set	Training Set	Test Set
Bicubic	369.42	358.67	71.73	358.21	71.84
SRCNN	422.82	390.79	78.14	389.46	78.09
WDSR	419.64	390.42	78.07	389.13	78.02
$\alpha = \frac{1}{7}$	425.52	392.80	78.55	391.60	78.51
$\alpha = \frac{1}{6}$	428.06	393.07	78.60	391.92	78.57
$\alpha = \frac{1}{5}$	430.90	394.12	78.81	392.82	78.76
$\alpha = \frac{1}{4}$	433.62	394.98	78.98	393.75	78.94
$\alpha = \frac{1}{3}$	**439.84**	**396.86**	**79.37**	**395.37**	**79.27**

method, the compressed datasets generated from results of our method have the largest file sizes. Furthermore, for our method, as the value of α increases, the sizes of the compressed datasets increase gradually. When α rises to 1/3, the compressed datasets have the largest sizes. Since the file size of a compressed dataset reflects the amount of high frequency contents in its corresponding reconstructed HR images, the results in Table IV demonstrate that our method can really recover more high frequency contents.

What's more, as a supplementary explanation, the amount of reconstructed high frequency information is not always positively related to PSNR. As shown in Table IV, SRCNN method generates approximately the same-sized compressed dataset as WDSR, but its PSNR is much lower than WDSR. This means that SRCNN recovers more inaccurate contents in HR image reconstruction, and also results in a lower classification accuracy in Table III. The same situation occurs when we test our new method with different parameter values. When α is set to 1/3, the network generates the largest compressed dataset. However, it results in a lower PSNR in super-resolution validation, as well as a lower accuracy in classification experiment. Therefore, it is important to maintain the balance between more high frequency information and a higher PSNR.

B. Spectrogram of the Super-Resolution Image

The Fourier Transform is a mathematical operation which can decompose an image into its sinusoidal and cosinoidal components, and it will output the frequency domain representation of this image. A spectrogram is a visual representation of the Fourier transformed image in which each point represents a particular frequency contained by the spatial domain image. If the DC values are not shifted to the center, the central part of the spectrogram is the high frequency and the peripheral part is the low frequency.

Figure 4 shows the spectrograms of the super-resolution images which are upscaled by different SR method. As shown by Figure 4(a), the bicubic interpolation method yields the lowest intensive values in central part of the spectrogram, it means that this method gains the least amount of high frequency contents. In addition, the original WDSR method constructs more high frequency contents than bicubic interpolation, and our method achieves the most high frequency contents among these methods.

The results shown in Figure 4 are consistent with the conclusion drawn from Table IV. Our super-resolution method, which corresponds to the largest compressed datasets generated by differential coding compression, also yields the highest intensive values in central part of the spectrogram. Hence, we can draw a conclusion that our method can recover more high frequency contents compared to the other state-of-the-art super-resolution methods.

VI. CONCLUSION

In this paper, we propose a new image super-resolution method in order to improve the classification performance for low-resolution images. The method introduces a weight map to denote positions of the pixels which contain more high frequency information in ground truth HR image. After that, during the training process, the network focuses on minimizing the difference between reconstructed HR image and ground truth HR image at these positions. The experiment results show that our method has two main advantages compared to the other state-of-the-art super-resolution methods. For one thing, it has been proven for two times that our method can really recover more high frequency contents in HR image reconstructing. For another, our method can better improve performance of the low-resolution image classification by scaling up the datasets using our method and retraining the network. Because of the similarities among tasks, our method can also be applied in image dehazing and denoising in future work.

VII. ACKNOWLEDGEMENTS

This work is supported by National Natural Science Foundation of China (61671332), China Scholarship Council (201806270244) and China Scholarship Council (201807990021).

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 779–788.

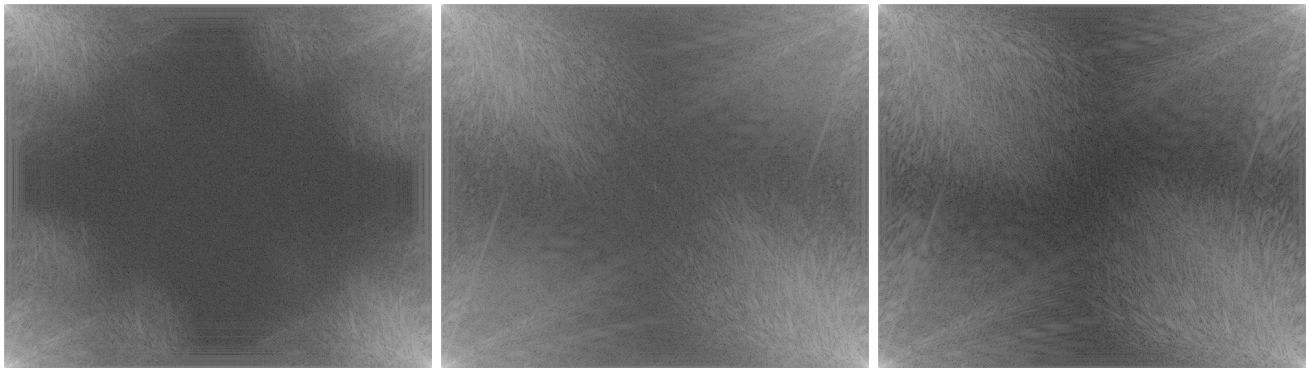
[3] X. Wu, D. Sahoo, and S. C. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, 2020.

[4] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

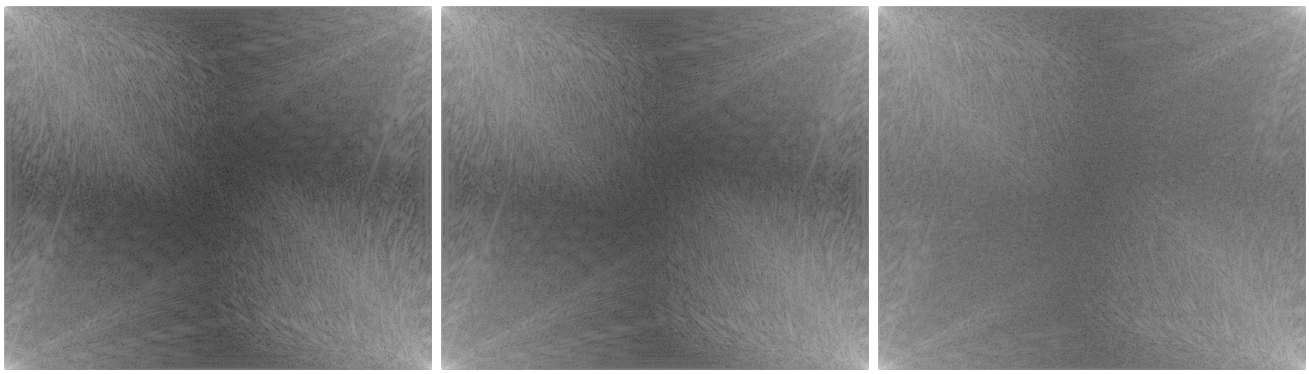
[5] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao, "Dehazenet: An end-to-end system for single image haze removal," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5187–5198, 2016.

[6] J. Qin, Y. Huang, and W. Wen, "Multi-scale feature fusion residual network for single image super-resolution," *Neurocomputing*, vol. 379, pp. 334 – 342, 2020.

[7] R. G. Vidal, S. Banerjee, K. Grm, V. Struc, and W. J. Scheirer, "Ugˆ 2: A video benchmark for assessing the impact of image restoration and enhancement on automatic visual recognition," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.



(a) Bicubic (b) SRCNN (c) WDSR



(d) $\alpha = \frac{1}{6}$ (e) $\alpha = \frac{1}{3}$ (f) Ground Truth

Fig. 4. Spectrograms of the images upscaled by bicubic, SRCNN, WDSR, our method. The DC values are not shifted to the center and the center parts represent the high frequency content. The intensity values of center parts increase from (a) to (f). The original LR image is cropped from the '0068×2.png' in training dataset of DIV2K.

[8] Y. Yuan, W. Yang, W. Ren, J. Liu, W. J. Scheirer, Z. Wang *et al.*, "Advancing image understanding in poor visibility environments: A collective benchmark study," *arXiv preprint arXiv:1904.04474*, 2019.

[9] W. S. Russell, "Polynomial interpolation schemes for internal derivative distributions on structured grids," *Applied Numerical Mathematics*, vol. 17, no. 2, pp. 129–171, 1995.

[10] J. Sun, Z. Xu, and H.-Y. Shum, "Image super-resolution using gradient profile prior," in *IEEE international conference on computer vision (CVPR)*, 2008.

[11] W. T. Freeman, E. C. Pasztor, and T. R. Jones, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.

[12] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *IEEE conference on computer vision and pattern recognition (CVPR) Workshops*, 2017, pp. 136–144.

[14] J. Yu, Y. Fan, J. Yang, N. Xu, Z. Wang, X. Wang, and T. Huang, "Wide activation for efficient and accurate image super-resolution," *arXiv preprint arXiv:1808.08718*, 2018.

[15] L. Zhou, Z. Wang, Y. Luo, and Z. Xiong, "Separability and compactness network for image recognition and superresolution," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3275–3286, 2019.

[16] 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)*, 2018, pp. 842–848.

[17] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on computational imaging*, vol. 3, no. 1, pp. 47–57, 2016.

[18] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017.

[19] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10 and cifar-100 datasets," https://www.cs.toronto.edu/ kriz/cifar.html.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[21] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in {*USENIX*} *Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] A. Horé and D. Ziou, "Image quality metrics: Psnr vs. ssim," in *2010 20th International Conference on Pattern Recognition*, 2010, pp. 2366–2369.

[24] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, 2013, pp. 1139–1147.

[25] Y.-Q. Shi and H. Sun, *Differential Coding*, 03 2019, pp. 59–80.

[26] L. D. Crocker, "Png: The portable network graphic format," *Dr Dobb's Journal-Software Tools for the Professional Programmer*, vol. 20, no. 7, pp. 36–45, 1995.