

METEOR: Measurable Energy Map Toward the Estimation of Resampling Rate via a Convolutional Neural Network

Feng Ding, Hanzhou Wu, Guopu Zhu, *Senior Member, IEEE* and Yun-Qing Shi, *Fellow, IEEE*

Abstract—In recent years, with the improvements in machine learning, image forensics has made considerable progress in detecting editing manipulations. This progress also raises more questions in image forensics research, such as can the parameters applied in a manipulation be estimated. Many parameter estimation works have already been performed. However, most of these works are based on mathematical analyses. In this paper, we attempt to solve a particular parameter estimation problem from a different aspect. Specifically, a new convolutional neural network (CNN) model is proposed to estimate the resampling rate for resampled images regardless of whether the image is upscaled or downsampled. This model features an original layer to generate a measurable energy map toward the estimation of resampling rate (METEOR). The METEOR layer is demonstrated to be an outstanding method that can assist in enhancing the estimation performance of the CNN. Furthermore, the METEOR layer can also increase the robustness of the CNN against JPEG compression, which makes it extremely important in realistic application scenarios. Our work has verified that machine learning, particularly CNNs, with proper optimization can also be refined to adapt to parameter estimation in digital forensics with excellent performance and robustness.

Index Terms—Image forensics, resampling, machine learning, convolutional neural network.

I. INTRODUCTION

INNOCENCE of images was lost long ago. Furthermore, the rapid development of image editing tools in recent decades has led to attacks on images becoming more frequent and more professional. A synthesized image with maliciously modified contents may tell a completely false story that could lead to immense harm to many people. Thus, fully understanding the editing history of an image can help us to understand the story behind an image more clearly. The concept of image forensics was proposed with the primary task of protecting the authenticity and integrity of images [1][2][3]. Many scientists have dedicated their research to defending the authenticity of images. In image forensics research, forgeries can be identified

This work was supported in part by the National Natural Science Foundation of China under Grant 61572489, Grant 61872350, and Grant 61902235, and in part by the Basic Research Program of Shenzhen under Grant JCYJ20170818163403748. (*Corresponding author: Guopu Zhu.*)

F. Ding is with the School of Medicine and Health Management, Tongji Medical College, Huazhong University of Science and Technology, Wuhan 430030, China, and also with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, GD 518055, China (E-mail: fding@albany.edu).

H. Wu is with Shanghai University, Shanghai 200444, China (E-mail: h.wu.phd@ieee.org).

G. Zhu is with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, GD 518055, China (E-mail: guopu.zhu@gmail.com).

Y. Shi is with the New Jersey Institute of Technology, NJ 07101, USA (E-mail: yun-qing.shi@njit.edu)

[4][5][6], hidden information can be revealed [7], and the editing history of images can be verified [8][9][10][11][12], and these contributions from forensics researchers guarantee the reliability of images.

With the development of artificial intelligence, researchers have applied machine learning methods [13] in the image forensics field. Frameworks based on a combination of hand-crafted feature extraction and supervised learning classification have succeeded in solving many forensics problems [14][15][16][17]. However, the handcrafted features have limitations. There are higher-order patterns that cannot be comprehended by the human brain. How to extract and adopt these deeper features remained a colossal challenge until the resurgence of neural networks in 2013. It is now the era of the convolutional neural network (CNN) [18]. CNNs are widely applied for many arduous tasks in various research fields, including image forensics.

Many forensics works have already been proposed based on CNNs [19]. Generally, these works share a similar objective, namely, determining whether a certain manipulation has been applied to images. Undoubtedly, all these works are important because every possible manipulation needs to be accurately identified to completely understand the image history. However, because of the development of new techniques, only identifying manipulations in image forensics can no longer satisfy the current circumstances. It can be foreseen that related higher requirements may be set, such as parameter estimation. Although not as many as detection works, some parameter estimation works have been performed [20][21], which is also our motivation for this work. Specifically, our work aims to employ a CNN to estimate the resampling rate rather than identify resampling.

Resampling [22] is one of the most commonly used image editing manipulations. This technique is primarily applied to adjust the sizes of images because resizing is one of the fundamental needs of many people. When discussing image resizing, in addition to resampling, there is also another well-known manipulation that needs to be mentioned: seam carving. Unlike resampling, seam carving is intended to be applied to downscale images by removing low-frequency components in the background. After the irrelevant background information is removed, the major objects are emphasized in the seam-carved images. In other words, the contents of images can be considered to be altered after being processed by seam carving. There is no doubt that seam-carving forensics [23][24][25][26] is quite important because decisive information may be found

in the background of images. In contrast, for resampling, all contents, including the background, in images persist and preserve the same ratio after being resampled, regardless of the intensity of frequency. This property makes resampling a preferred option for adjusting the sizes of images with the contents remaining untouched. Furthermore, resampling can not only shrink images but also enlarge them without distortion. This makes resampling a premium tool for the general purpose of image scaling.

In practice, a small image is always upscaled to be viewed by human eyes from a distance, whereas a large, blurry image may be downsampled to provide a clear view with a higher resolution. In addition, it is also highly possible that only a certain part of the image could be scaled under various motivations. For instance, in the tampering of a copy-paste attack [27], the tampered parts are most likely to be resampled prior to pasting to accommodate the background scene. Considering the importance of tampering detection, it can be guaranteed that the confirmation of resampled parts plays an essential role in image forensics. Several remarkable works for resampling forensics have been reported [28][29][30].

For manipulation by resampling, the most important parameter is the resampling rate. This parameter generally controls the size of a resampled image. Determining the resampling rate enables us to estimate the image editing history more thoroughly. Meanwhile, for tampering detection, it is also important for us to trace the source of identified tampered parts. Thus, as mentioned above, the resampling rate is clearly a crucial element for solving this problem. With an estimated resampling rate, we are capable of restoring the resampled parts to their original size, which makes it simple to trace back to the original source.

As mentioned above, there are already published works that attempted to estimate the parameters of manipulations. However, most of these works are based on mathematical analyses. Machine learning and neural networks are convincing tools for detecting manipulations in digital forensics; however, these tools are not preferred for estimating parameters. Therefore, in this paper, we want to explore the potential of CNNs. An original scheme based on a CNN is proposed here to discriminate the limited diversity in resampled images with different resampling rates. Our CNN model is built based on certain principal and observed experimental results, which will also be investigated in this paper. The proposed model is equipped with a newly designed layer called the 'METEOR' layer to analyze the resampled images from a new perspective. This layer is adopted to analyze the energy information of images. The image residuals after processing by the METEOR layer are regarded as energy feature maps that serve as decisive components to enhance the performance of the CNN. In addition, the deviation caused by different contents in images can be constrained by the METEOR layer to enhance the convergence performance of the CNN. The METEOR layer enables the CNN to be a more reliable tool for evaluating the resampling rate. The superiority of the proposed approach is justified via our simulations.

To summarize the above, the main contributions of this paper are as follows:

- 1) A CNN model is proposed to serve as an estimator of the resampling rate for resampled images. It can help us reach a step further in evaluating image editing history [31][32]. Moreover, with resampling rate, it also enables us to trace the original source, especially for identifying tampered contents [33][34].
- 2) A new layer focusing on analyzing image energy, named 'METEOR', is introduced. This layer aims to refine the performance of the CNN model in estimating the resampling rate from different aspects. It is also capable of updating its kernel size to enhance the estimation performance of the proposed model. The energy feature map effectively increases the estimation accuracy of the CNN. The METEOR layer can also offer stability for the convergence of the CNN. Furthermore, the irrelevant textures and other information can be eliminated from images after processing by METEOR, which restrains the deviation caused by image contents. Experiments are conducted to investigate the capabilities of METEOR.
- 3) The method is based on deep learning. This demonstrates that machine learning methods such as CNNs are also capable of estimating parameters for image forensics. Furthermore, the principles for building our CNN architecture, such as the dimension, pooling strategy and choice of activation function, to solve image forensics problems are investigated to provide additional value. These components are the key elements in building CNN models.

The remainder of this paper is organized as follows. In the next section, we discuss several novel works in resampling forensics. In Section III, we briefly introduce the theory of resampling. The proposed scheme is described in detail in Section IV. The assessment of the proposed scheme based on experiments is discussed in V. Finally, the conclusion is presented.

II. RELATED WORKS IN RESAMPLING FORENSICS

Most digital forensics researchers recognize that the paper published in 2005 by Popescu and Farid initiated image forensics research against resampling [28]. They proposed a method that employs the expectation-maximization (EM) algorithm to measure the correlation of pixels to determine whether images have been resampled. Their method can achieve excellent detection accuracy, which is higher than 90% in most resampling cases. However, the EM algorithm is considered to be a complicated, computationally demanding method that consumes a great deal of time. In addition, the detection performance for downsampling is relatively poor. Later, in 2008, this work was followed by that of Kirchner. He proposed a new detector with an algorithm other than an EM algorithm [35]. His method is based on analyzing the gradient of the p-map spectrum, which is not only easier to implement but also computationally efficient. The detection performance is also slightly enhanced for downsampling.

Afterwards, researchers started forensics research with machine learning algorithms. In the early stage of machine-learning-based forensics, researchers designed handcrafted features and fed these features into supervised machine learning models for training. A well-trained classifier is capable of discriminating images with designated features. This machine learning procedure was adopted to solve many forensics problems and to detect resampling. Among them, one of the most representative methods was proposed in 2012 by Feng *et al.* [29]. They discovered alternation of the energy density property and extracted it via a feature with only 19 dimensions. The combination of energy density and support vector machine (SVM) has been proven to be a more advanced tool, in which the detection performance was enhanced to higher than 95%, even when the images are downsampled.

In recent years, most machine learning scientists have switched to focusing on neural networks again. CNNs have also been introduced in the forensics research area. Bayar and Stamm employed a CNN to detect resampling, especially resampling applied before the images were compressed [30]. Although the traces of resampling may fade after compression, it is still not difficult for their CNN model to identify manipulation by resampling in most cases. Later, the same group proposed the constrained CNN model [36], which is a universal tool that can detect resampling and many other manipulations. Given the experimental results, the detection performance of constrained CNN is impressive.

Apart from the above detection works, a few algorithms based on mathematical model analyses have been proposed for estimating the scaling factors under certain conditions. Vázquez *et al.* [37] adopted a random matrix to measure the maximum likelihood for images that are upscaled. Liu *et al.* [38] employed a histogram of difference image extremum intervals to estimate the scaling factors in downsampled pre-compressed images. Their method has been shown to achieve impressive estimation performance in relative particular cases.

Thus far, in image forensics, it appears that machine learning methods can only be employed as binary classifiers to qualify as detectors. In contrast, most parameter estimation works are realized by mathematical analyses. The machine learning methods appear to be inappropriate for use as estimators. It is a challenge for us to use a CNN as an estimator.

III. RESAMPLING

Resampling in image processing is generally categorized as upsampling and downsampling. Upsampling increases the sampling rate of a signal, while downsampling decreases the sampling rate. In other words, the size of the image increases after upsampling and decreases after downsampling. Regardless of the change in size, both upsampling and downsampling can be achieved with the same operation: interpolation. In image processing, interpolation [39] is a procedure for creating new pixels. The intensity of a new pixel is often the sum of weighted pixels in the neighborhood. The weight factor is highly related to the distances between interpolated pixels and neighboring pixels.

There are 3 common interpolation algorithms, which are designed to satisfy different purposes [40]. Nearest neighbor

interpolation is the simplest method, which is computationally efficient. If image quality is the major concern for resampled images, 'bicubic' interpolation is recommended. This approach [41] can be regarded as applying convolution with the bicubic interpolator to the original image. In addition to these two approaches, there is also 'bilinear' interpolation, which is a tradeoff between computational speed and image quality that is appropriate for most cases.

Other interpolation methods also exist that can be applied to adjust the sizes of images. For example, 'Lanczos' can be regarded as an alternative version of 'bicubic', which is also a popular interpolation strategy. The difference between these two methods is that during interpolation, 'Lanczos' replaces the bicubic kernel with the Lanczos kernel for convolution. Hence, similar to 'bicubic', a high image quality is achieved at the cost of a heavier computational burden.

IV. PROPOSED METHOD

A. Convolutional neural network

Deep learning [42] has been a very popular topic among scientists since 2013. As the most representative deep learning method, a CNN with a feed-forward structure is ideally suited for analyzing input data. In fact, the CNNs that we currently consider are refined versions that are capable of analyzing a substantial quantity of data with the assistance of GPU implementations. Through automatic learning from many samples via backward propagation during training, the CNN is able to distinguish the learned features. This property is ideal for analyzing visual imagery [43][44]. Thus, CNNs are mostly applied by scientists in the fields of computer vision and image forensics to handle images.

A CNN model consists of layers of different categories. The input layer, which is the entrance for input data, is called the 'data layer', while the exit of the entire network is always set as the 'loss layer' to evaluate the performance of the network by outputting the classification accuracy, training loss and so on. Typically, a CNN is composed of one data layer for input, one loss layer as output and multiple hidden layers for the learning procedure. The hidden layers are always considered the trunk or central nervous system for the neural network, which is the most important part. These layers consist of vision layers, normalization layers and activation layers. The structure of the proposed CNN model in our work is illustrated in Fig. 1.

At the bottom and top of our model, similar to most models proposed by other researchers, the data layer and loss layer are set as usual. For the data layer, we choose the 'image data layer' to directly input the raw images. For output, the 'softmax' layer and accuracy layer are selected to output the training loss and classification accuracy to assess the model.

The newly designed METEOR layer, which will be described in detail in the following subsection, follows the data layer. This layer is adopted to guide the network to analyze the energy features of images.

Between the METEOR layer and loss layer, there are hidden layers built with convolutional, normalization, fully connected and pooling layers. As shown in Fig. 1, multiple groups of repeated convolutional layers, batch normalization layers,

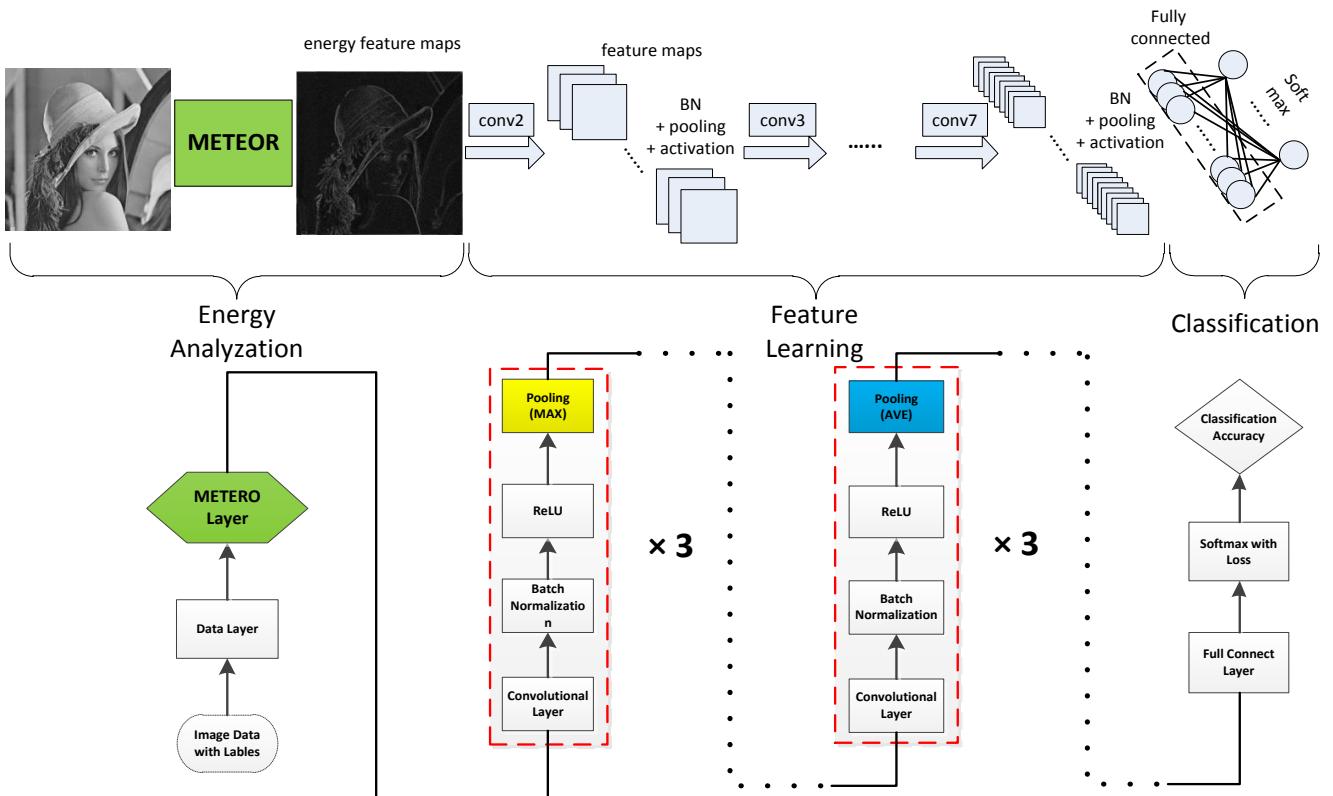


Fig. 1: Proposed CNN model.

activation layers and pooling layers appear periodically. These layers can be described as vision groups. A higher-dimensional CNN architecture with more vision groups is generally considered deeper because the learning ability is relatively stronger. The dimension of the CNN must be carefully set for each specific problem to ensure that the model properly fits the problem. In our model, the dimension is set to 6, which we will discuss in the next section.

The other factors that have major impacts on the CNN model are the pooling and activation layers. The pooling layer decreases the feature dimension by combining the outputs from neuron clusters in the prior layer. This operation is useful to prevent overfitting and to enhance computational efficiency. A pooling method can be normally chosen from either 'maximize' or 'average'. The activation layer is adopted to increase the nonlinear property for making decisions during classification. It is particularly effective for solving the multilabel classification problem. The activation functions most preferred in recent years are 'TanH' and 'ReLU'. In our model, we choose 'maximize' for the first 3 pooling layers and 'average' for the last 3 pooling layers; meanwhile, 'ReLU' is chosen as the function for all activation layers. These choices are also based on our experiments. Related discussions can also be found in Section V.

B. Measurable energy toward the evaluation of resampling rate

Our proposed CNN structure is simple and does not have complicated modifications. It is similar to many traditional models proposed in computer vision areas. Similar architectures are employed to address topics such as handwriting recognition, object recognition, face identification, gesture identification and so on. The advantages of self-learning, backwards propagation and local feature extraction make CNNs the state-of-the-art for many problems. However, we want to employ a CNN to serve as an estimator in image forensics, which is another novel application for CNNs. Certain necessary modifications need to be made to adapt the CNN for estimating the resampling rate.

The most important part for the proposed method to achieve success is for the model to be able to accurately make a diagnosis of resampled images based on the essence of resampling. The resampling can be achieved via the interpolation methods introduced in the last section. Thus, for such manipulations, each pixel in the resampled images is an interpolated pixel, whose value is the sum of weighted neighboring pixels in original images. In most cases, the value of an interpolated pixel should be close to that of its neighbors in the original images. In other words, interpolated pixels tend to be assigned with similar intensity in local areas because they share many homogeneous neighbors. Hence, for upscaled images, with more pixels interpolated in the same local area, more regions with lower gradients can be located. Consequently, the image

tends to possess lower energy. Symmetrically, for downscaled images, because fewer pixels are interpolated in local areas, the value difference of interpolated pixels is amplified. Consequently, more pixels with higher gradients can be found that lead to higher energy for downscaled images. To capture such alternations in pixel gradients via machine learning, a feature extractor that can provide scalable energy information should be created. Additionally, our model is built based on a CNN. As mentioned above, CNNs were originally created to solve computer vision problems. It is inevitable that the content of images could impact a CNN model in making decisions. For our problem of estimating the resampling rate, such an impact is undesirable.

Summarizing the above discussions, to assist the CNN in serving as an estimator, the ideal original layer is designed with the following principles. It can provide scalable energy information to be analyzed by the CNN for resampling rate estimation. Meanwhile, it can also restrict the impact brought by image contents to avoid unnecessary deviations. Therefore, the layer is designed as follows.

In the first step, a kernel is composed based on a frame of an $m \times m$ block that contains m^2 elements. m indicates the size of the kernel. Note that a CNN has an automatic learning ability that is realized by backward propagation. Thus, the kernel size is dynamic and can be updated during training with stochastic gradient descent. To make the kernel a convenient tool, m is strictly limited to odd positive integers between 3 and 9. The variable m enables the CNN model to adjust its kernel size to adapt to images resampled with different factors. This function is crucial because the estimation is blind for most cases. The resampled image is given without awareness of whether it is upscaled or downscaled.

Regarding the theory of interpolation, in the kernel, the weight value for each element should only be relevant to the geometry distance, which is the direct shortest distance to the element in the center. The closer the distance is, the heavier is the weight for the element. The weight w_n for each element in the kernel is defined as

$$w_n = \frac{1}{\sqrt{(x_n - x_c)^2 + (y_n - y_c)^2}}, \quad (1)$$

Note that the weight coefficient should be normalized by

$$W_n = \frac{w_n}{\sum_{n=0}^{m^2-1} w_n}, \quad (2)$$

where x_c and y_c are the coordinates of center element M_c in kernel, while x_n and y_n indicate the coordinates for the neighbor elements M_n around center element M_c .

After the weight coefficients are determined, the kernel is completed with the W_n assigned to elements in relative position, while the weight in the center W_c is fixed to be 0. Afterwards, a new pixel P_t can be generated by convolution with an image block I ,

$$\begin{aligned} P_t &= \sum_{n=0}^{m^2-2} W_n P_I + W_c P_c \\ &= \sum_{n=0}^{m^2-2} W_n P_I \end{aligned} \quad (3)$$

where P_c is the pixel in the center of I , P_I represents the surrounding pixels to P_c in I . Recall that the kernel size m is a variable integer. Here, in the following equation, the size of I is set to be strictly equal to m to match the kernel size.

During our simulations, we also observed that the variable kernel size may introduce errors. Hence, inspired by the 'GoogLeNet' [45], a flexible padding strategy is applied to fix the kernel size. To avoid confusion, the kernel after padding is defined as the core in the remainder of this paper. Under this strategy, the core is fixed to 9. If m is less than 9, we use zeros as padding around the determined elements. The null padding tends to create necessary extra columns and rows to ensure that the core size is equal to 9. Meanwhile, the weights with nonzero values are secured in the kernel which is the center part of core. The core of METEOR is depicted in Fig. 2.

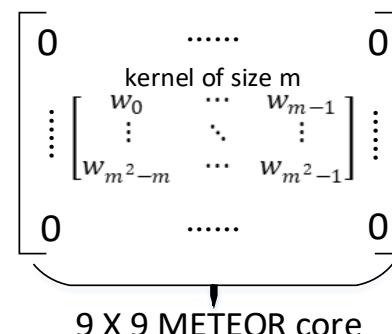


Fig. 2: The standard METEOR core with zero padding.

After convolution with the core of METEOR, post-core processing is necessary to derive the energy feature map. The calculated P_t and P_c are processed to derive e by

$$e = \sqrt{|(P_t)^2 - (P_c)^2|} \quad (4)$$

where P_c is the center pixel in image block I . The derived e can be considered as the energy for the center pixel. It can also be regarded as a quantized gradient of P_c , which is highly relevant to the image energy. The value of e is assigned to P_c as the value of P_c in the feature map. After e for each pixel is determined, a new feature map E can be created with all calculated E substituting the relative original pixels as the residual of the original image. E is the desired energy feature map. This feature map can be utilized as the input for the following layers. Technically, energy feature maps with similar patterns of measurable energy are categorized in the same group. Thus, the estimation can be achieved. Because this unique feature map is designed to estimate the resampling rate, it is named 'METEOR' (measurable energy towards the estimation of resampling rate). A sample energy map is displayed in Fig. 3. The entire process for METEOR is given in Algorithm 1.

From the sample energy map, it can be observed that silhouettes of the portrait are well preserved, while the textures in the images are mostly removed. For this result, there are several points that must be discussed. First, the edges with higher contrast in images are always considered to be the parts that



Fig. 3: (a) Original Lena image; (b) image residual after processing by METEOR ($m = 5$).

Algorithm 1 Procedure of METEOR layer.

```

1: Initialize the kernel (center parts) with  $m = 5$ 
2: while  $1 \leq i \leq MaxIteration$  do
3:   Assign the weights to the elements in the kernel
4:   if  $m < 9$  then
5:     Do zero padding to ensure the core size is 9
6:   end if
7:   Convolve core with input image to calculate  $P_t$ ,
   post-process to derive energy feature map  $E$ 
8: end while
9: Forward pass to have the energy feature map processed
10: Backward pass to update the parameters for METEOR
11: if The model converges then
12:   Exit
13: end if
14: End

```

possess higher frequency components. Human eyes are sensitive to the high contrast areas, which is also the reason why contrast enhancement is commonly applied to improve image quality for human vision systems. The silhouettes displayed in Fig. 3 are so clear that they may lead to confusion about the energy feature map being a high-frequency residual image; consequently, METEOR may be considered nothing more than a high-pass filter. However, METEOR actually delivers measurable energy information regardless of the frequency intensity. If observed carefully, many subtle low-frequency elements can also be found in addition to high-frequency components. Second, many scientists are trying to understand the theory behind CNNs; regarding recent works [46], we realized that CNNs bias textures over silhouettes or shapes for classification tasks. Unlike the human eyes that prefer to recognize objects by shapes and colors, neural networks are

more sensitive to the local features or texture of objects. Thus, with the textures filtered out in the energy map, the METEOR layer makes the object recognition nearly impossible for the CNN. Thus, our initial motivation to design an assistance tool that is capable of tracing gradient information and restricting the compact of image content can be satisfied simultaneously with the METEOR layer. Our remaining task is to assess the evaluation performance of the proposed model.

V. EXPERIMENTAL RESULTS AND DISCUSSION

A. Comparison with prior works

We choose the BOSS raw dataset [47] and RAISE [48] as our image databases for experiments because they can deliver raw images in high resolution with a large size. The raw images are taken by different cameras under different lighting conditions and scenes, which makes these ideal datasets for image-based research works. In addition, both datasets contain large numbers of images. The data size is a major concern for deep learning research. It is well known that more data bring promising results in scientific research.

For the parameters of the CNN, the learning rate varies for different experiments; however, for most cases, it is set between 0.005 and 0.01. The momentum is fixed to 0.9. Most strides for the convolutional layer are 1, and the strides for the pooling layers are 2. The default kernel size m is set to 5 for all the following experiments. All experiments are conducted on a computer equipped with GTX-1080ti GPU. Besides, all the images are pre-processed with necessary manipulations such as cropping, RGB to grayscale conversion and so on. The input images to the network are gray level images that fixed to size of 512×512 in all experiments.

For the first step, we test our method to determine whether the proposed method is capable of identifying the manipulation of resampling under different resampling rates. Note that, the

proposed model tend to serve as an estimator for resampling rate, thus, the detection of resampling is only an additional function. In this case, the classifier makes a binary decision of whether the given image is resampled. Comparisons are made with Feng *et al.*'s method [29], Bayar *et al.*'s method [30] [36]. As introduced in Section III, [29] is the best known method for resampling forensics based on SVM. [30] is the latest method for resampling forensics based on a CNN. [36] is a universal forensics tool that is designed to detect multiple manipulations and is thus far the most powerful detection tool. In the first experiment, the resampled images are interpolated with different rates via a random method. 'Mixed' indicates the images with different resampling rates, and the interpolation methods are mixed up for the test. The result is shown in Table I. As shown in the table, all methods can achieve excellent accuracy on both datasets. Even when the resampled images with different resampling rates are mixed together, all methods can still achieve an accuracy over 93%. This result verifies that the identification of resampling appears to no longer be a challenge in image forensics. The detection for downsampling was once a challenge in image forensics [28]; however, with the development of new techniques, this task is now much easier. Furthermore, it can be observed that the methods based on CNNs outperform the combination of handcrafted features and generalized linear classifiers. All three CNNs have achieved exceptional detection accuracy with similar performance for detecting resampling. This is not a coincidence because the superiority of CNNs has been verified many times in various fields. Our experiment again proves this point.

Afterwards, an experiment for evaluating the resampling rate is conducted because the major purpose of this paper is to estimate the resampling rate in the resampled images. To perform a fair comparison, although there are other mathematical models that aim to estimate the resampling rate, as referred to in Section II, only the methods based on machine learning are employed in this experiment. Hence, it can be considered as a problem for multiple label classification. Therefore, in this experiment, the resampling rates range from 0.2 to 2 with 0.1 as interval are applied in images. All 4 popular interpolation methods introduced in Section III are tested respectively in this experiment. In addition, the original images are also included and labeled as resampling rate 1 for this experiment. All images are labeled and wrapped together and serve as the input data for training and validation. Note that although linear classifiers such as SVM are well known for their binary classification capability, they can also be trained to address multilabel classification problems. Hence, to approximately simulate the resampling rate estimation for [29], we select resampled images with the same resampling rate as the target group and then compulsively combine the remaining images of different resampling rates as the control group. Then, we can test the classification performance of [29] under various interpolation methods. The comparison results are reported in Table II. The percentage number indicates the ratio of correct prediction over the entire test set. In addition, we also include the confusion matrix based on the estimation for 'bicubic' resampled RAISE images with our proposed model as shown

in Table III. It can provide an explicit explanation how the estimation can be achieved via classification.

As depicted in Table II, at first, the neural networks, even with only the preset layers, can somehow predict the resampling rate with an acceptable accuracy. However, the algorithm based on a linear classifier fails to fulfill the task. The classification accuracy for distinguishing resampled images with a single resampling rate from the other resampled images is near 50%, which suggests that the SVM classifies the given data arbitrarily. To differentiate the arbitrary guess of SVM from the actual estimation accuracy provided in CNN models, it is recorded as 'NA' for 'Not Available' in the table. Second, although Bayar *et al.*'s methods [30] and [36] can still archive an estimation accuracy of over 90% for most cases, our proposed CNN model with the METEOR layer can reach an estimation of approximately 96-97% regardless of the interpolation methods or datasets in this experiment. It is quite easy for the proposed model to outperform [30] and [36] by at least 4% in each case. At this high baseline, an improvement of 4% could be considered as significant. Although the constrained neural network is a premium tool to identify most image manipulations, the METEOR layer is a more advanced tool for estimating the resampling rate.

Summarized from the all experiments above, our proposed model is verified to be the most suitable method with the best estimation accuracy for analyzing the manipulation of resampling. In addition to the identification of resampling, the most important result is that the proposed model can successfully serve as an estimator of resampling rate.

B. Evaluation for the METEOR layer

Another important purpose of our experiment is to investigate the capability of the original METEOR layer because it is one of the major novelties of the proposed method. Here, we conduct experiments to evaluate this layer from different aspects. Because the METEOR layer produces image residuals as an energy feature map, we can simply remove it from the proposed model to skip this step. The model without the METEOR layer is an ideal control target for comparison with our proposed model to evaluate the performance of METEOR. Hence, we employ both models to estimate the resampling rate for images resampled by different interpolation algorithms. The resampling rates applied to evaluate METEOR layer is also from 0.2 to 2 with 0.1 as interval. We also test on each interpolation methods respectively on both datasets. The comparison results for estimation accuracy can be found in Table IV.

From the comparison, it is quite clear that the newly designed METEOR layer is the decisive element for estimating the resampling rate. When removing this layer, the estimation accuracy decreases for different interpolation methods. For both datasets, the METEOR layer can improve the estimation accuracy by at least 6%. Given the results, it can easily be concluded that the METEOR layer refines the CNN model in improving estimation accuracy. Moreover, we also observed that our model, even without METEOR, appears to be more suitable for estimating the resampling rate than the model

TABLE I: Detection accuracy for resampling under different rates on both datasets.

Rate	BOSS				RAISE			
	[29]	[30]	[36]	Our model	[29]	[30]	[36]	Our model
0.5	0.9637	0.9825	0.9915	0.9904	0.9712	0.9757	0.9869	0.9847
0.75	0.9585	0.9878	0.9902	0.9875	0.9715	0.9925	0.9877	0.9936
1.25	0.9733	0.9862	0.9865	0.9925	0.9841	0.9893	0.9931	0.9927
1.50	0.9617	0.9814	0.9877	0.9883	0.9792	0.9936	0.9944	0.9921
1.75	0.9755	0.9895	0.9925	0.9912	0.9823	0.9942	0.9950	0.9977
2.00	0.9728	0.9927	0.9911	0.9939	0.9866	0.9965	0.9939	0.9950
mixed	0.9340	0.9689	0.9823	0.9854	0.9317	0.9716	0.9921	0.9900

TABLE II: Estimation performance comparison for different interpolation methods on both datasets.

Method	BOSS				RAISE			
	[29]	[30]	[36]	Our model	[29]	[30]	[36]	Our model
Nearest	NA	0.8875	0.9044	0.9767	NA	0.8935	0.8981	0.9705
Bilinear	NA	0.8765	0.8912	0.9653	NA	0.9007	0.9133	0.9642
Bicubic	NA	0.9022	0.9381	0.9725	NA	0.9078	0.9181	0.9694
Lanczos	NA	0.8916	0.9265	0.9730	NA	0.8850	0.9116	0.9796

TABLE III: Confusion matrix for estimation on 'bicubic' interpolated RAISE images, ground truth (rows) versus prediction (columns)

rate	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
0.2	.9173	.0624	.0186	.0002	.0006	.0002	.0000	.0005	.0000	.0000	.0000	.0001	.0000	.0000	.0000	.0000	.0000	.0000	.0000
0.3	.0436	.9235	.0320	.0002	.0000	.0005	.0000	.0000	.0000	.0001	.0000	.0000	.0000	.0000	.0000	.0001	.0000	.0000	.0000
0.4	.0024	.0221	.9527	.0206	.0007	.0000	.0000	.0009	.0000	.0000	.0002	.0000	.0000	.0000	.0002	.0001	.0000	.0000	.0000
0.5	.0001	.0001	.0327	.9487	.0179	.0000	.0000	.0001	.0000	.0002	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
0.6	.0000	.0001	.0056	.0156	.9613	.0141	.0028	.0000	.0000	.0002	.0000	.0000	.0000	.0000	.0000	.0001	.0000	.0000	.0000
0.7	.0000	.0000	.0000	.0037	.0067	.9795	.0060	.0029	.0004	.0005	.0000	.0000	.0000	.0000	.0002	.0000	.0000	.0000	.0000
0.8	.0000	.0000	.0005	.0001	.0060	.0109	.9638	.0147	.0005	.0022	.0009	.0000	.0000	.0000	.0000	.0001	.0001	.0000	.0000
0.9	.0000	.0002	.0000	.0000	.0031	.0029	.0121	.9572	.0156	.0077	.0010	.0000	.0000	.0001	.0000	.0000	.0000	.0000	.0000
1.0	.0000	.0000	.0001	.0000	.0001	.0000	.0037	.0102	.9701	.0090	.0054	.0002	.0010	.0000	.0001	.0000	.0000	.0000	.0001
1.1	.0000	.0000	.0000	.0000	.0002	.0000	.0000	.0044	.0056	.9811	.0077	.0002	.0004	.0000	.0000	.0001	.0000	.0000	.0001
1.2	.0000	.0000	.0000	.0001	.0001	.0000	.0001	.0000	.0001	.0058	.9884	.0051	.0001	.0000	.0000	.0001	.0000	.0000	.0001
1.3	.0000	.0001	.0000	.0000	.0000	.0000	.0001	.0000	.0002	.0036	.9930	.0031	.0000	.0001	.0000	.0000	.0000	.0000	.0000
1.4	.0000	.0000	.0001	.0000	.0001	.0000	.0000	.0000	.0000	.0005	.0114	.9796	.0071	.0002	.0009	.0000	.0000	.0000	.0000
1.5	.0000	.0000	.0000	.0001	.0000	.0000	.0001	.0000	.0000	.0007	.0001	.0036	.0026	.00876	.0050	.0002	.0000	.0001	.0000
1.6	.0000	.0000	.0001	.0000	.0000	.0000	.0000	.0001	.0000	.0000	.0001	.0000	.0023	.0012	.9915	.0026	.0017	.0000	.0002
1.7	.0000	.0000	.0000	.0001	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0001	.0006	.0010	.9971	.0009	.0002	.0000	.0000
1.8	.0001	.0000	.0000	.0001	.0000	.0000	.0000	.0000	.0001	.0000	.0000	.0001	.0002	.0002	.0042	.0037	.9864	.0015	.0033
1.9	.0000	.0001	.0000	.0000	.0000	.0000	.0001	.0000	.0000	.0000	.0000	.0001	.0001	.0000	.0021	.0036	.9800	.0140	.0000
2.0	.0000	.0000	.0002	.0000	.0000	.0002	.0000	.0001	.0000	.0001	.0000	.0001	.0001	.0000	.0009	.0027	.0356	.9598	.0000

TABLE IV: Comparison of models with and without METEOR on both datasets.

Model	BOSS				RAISE			
	Nearest	Bilinear	Bicubic	Lanczos	Nearest	Bilinear	Bicubic	Lanczos
CNN w/out METEOR	0.9012	0.8871	0.9103	0.9110	0.8976	0.8985	0.9034	0.9048
CNN w/ METEOR	0.9767	0.9653	0.9725	0.9730	0.9705	0.9642	0.9694	0.9796

proposed by Bayar *et al.* [30]. The relevant principle for building our own CNN architecture will be discussed later.

During our experiment, we also observed that, apart from the estimation accuracy, the METEOR layer also enhances the performance of the CNN from another aspect: the convergence of the model. A proper approach to evaluate the convergence behavior is via the accuracy based on epochs. The 'bicubic' resampled images are chosen here. For a clear comparison, the convergence performances of the two models on different

datasets are illustrated in Fig. 4 and Fig. 5, respectively.

As shown in Fig. 4, the CNN can reach convergence earlier if METEOR is applied. It only takes approximately 7 epochs for convergence with METEOR, compared to approximately 13 epochs without METEOR. The METEOR layer has a positive effect on the CNN model because the convergence is significantly advanced on counts of epochs. A similar phenomenon can also be observed on the RAISE dataset in Fig. 5.

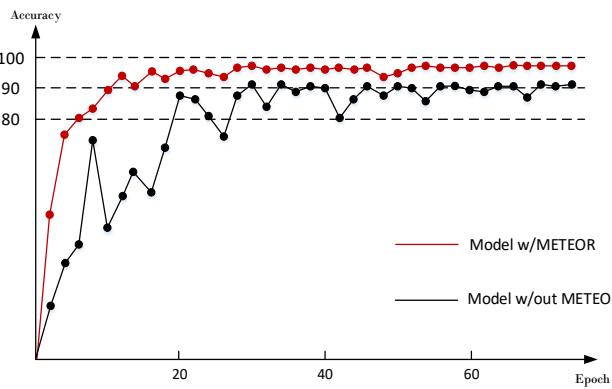


Fig. 4: Convergence performance of the proposed CNN model with and without METEOR for bicubic interpolation on BOSS.

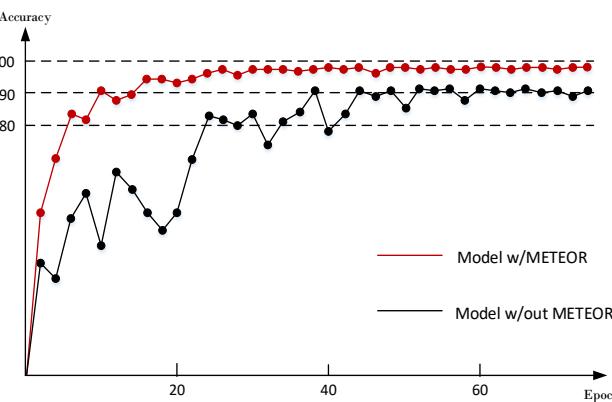


Fig. 5: Convergence performance of the proposed CNN model with and without METEOR for bicubic interpolation on RAISE.

In addition, convergence of model without METEOR also lacks stability, and oscillations can be found in the estimation accuracy as well as loss function. This result may be caused by the ambiguity of data, as neural networks are quite sensitive to the dataset. Because the training set and validation set are randomly selected from the dataset, oscillations are inevitable because images with fewer details may be chosen for the experiment. These images with less information will undoubtedly have a negative impact on the convergence of the CNN model. In particular, as shown in Fig. 5 for the RAISE dataset, the oscillations are so severe that even after 30 epochs, the model cannot converge with good results. However, with METEOR, such oscillations can also be constrained. The METEOR layer provides an impressive effect whereby the model reaches convergence quickly and solidly. The refined energy information can precisely capture the difference of traces left by different resampling rates. This development again demonstrates the superiority of the proposed METEOR layer.

Finally, a simulation to test the robustness of the METEOR layer against compressed resampled images is conducted. Compressed image formats are popular because they achieve a decent balance between image size and quality. Among all image compression algorithms, there is no doubt that 'JPEG' is the most common format for storing digital images. For this

experiment, we choose 'bicubic' as the interpolation method because it can deliver resampled images with the relatively high quality among all interpolation methods. The information loss and distortion should be the most severe for 'bicubic' resampled images after comparison, which leads to the most challenging case. The selected images are compressed via the 'JPEG' algorithm with quality factors of 50 and 75 as the dataset. During compression, certain image information is dropped, and the estimation of the resampling rate may be more difficult in our hypothesis. The experimental result is displayed in Table V.

TABLE V: Estimation accuracy for models with and without METEOR against JPEG-compressed images for both datasets.

Model	BOSS		RAISE	
	$Q = 50$	$Q = 75$	$Q = 50$	$Q = 75$
CNN w/out METEOR	0.5932	0.6270	0.6355	0.7159
CNN w/ METEOR	0.8976	0.9288	0.9059	0.9415

The result explicitly explains how important METEOR is for the CNN model. The model without METEOR fails to evaluate the resampling rate, and the estimation accuracy is poor. The information loss is critical in JPEG images; thus, the model without METEOR cannot handle the compressed images. Surprisingly, under such conditions, the same model with the METEOR layer can still estimate the resampling rate from the distorted energy information, which can be categorized as incredible. Even for the heavily compressed images with $Q = 50$, it can still reach an accuracy near 90%. Considering how much 'JPEG' is employed in our daily lives, the tremendous advantage of the METEOR layer against JPEG-compressed images makes it very valuable.

In summary, the METEOR layer can improve the estimation accuracy and enhance the overall convergence performance of the model; it is also robust against compressed images. The METEOR layer fulfills its responsibility as designed.

C. Discussion of CNN architecture design

The CNN model is a subtle complex that consists of layers with different choices. In addition, there are so many parameters and hyperparameters that are configurable for refining the model for different problems. In most cases, to solve a scientific problem, a proper network is built to be validated as effective. Afterwards, the effective network is fine-tuned by optimizing other hyperparameters. Hence, determining the key elements, such as the dimension of the network, the activation functions and the pooling methods, is the main priority.

During the procedure where we built the framework of our own model, we realized that it is necessary to discuss the architecture. Our main goal when building the CNN model is to keep it simple but effective. All unnecessary parts or layers are not considered to make it easy to realize. Overall, the essence of research via CNNs is to build reliable models to solve problems but not architectures with elegant shapes. Here, we conducted several experiments concentrating on the crucial components of CNN models to discuss the logic for

building our own model for the estimation of resampling rate. The other point needs to be mentioned is that we would like to deliver the instructions for building CNNs as exactly what we have done for our experiments. In the initial stage, instead testing models with all possible labels and images, we test models with reduced amount of data by applying less resampling labels. Therefore, for all the experiments in this subsection, the images are randomly resampled with equal probability of rates [0.50, 0.75, 1.25, 1.50, 1.75, 2.00]. Other than that, in order to create higher diversity for data, a total of 10000 images were randomly drawn from BOSS and RAISE to form a new dataset for the following experiments. There is no guarantee that the models tested with these data should be the absolutely best model with hyper-parameters fine-tuned in perfect status. However, it should be the most advanced strategy for building a proper CNN model considering the trade-off of time consumption and performance optimization.

At first, it is preferred to evaluate the capability of CNNs with different dimensions to determine the depth of the model. Generally, as referred to in Section IV, the output of a vision group is feature maps. These feature maps are employed as input for the next vision group to extract deeper feature maps with a smaller size but larger amount. A proper dimension can achieve the tradeoff between classification accuracy and computational efficiency. Therefore, for different problems or different purposes, the moderate dimension of the CNN varies. We conducted experiments with different CNN dimensions D and with different pooling strategies and activation functions against the evaluation of 'bicubic' interpolation. First, the results for models with different D are reported in Table VI and Fig. 6.

TABLE VI: Classification accuracy for models with different dimensions

Depth of dimension	Estimation accuracy
$D = 4$	94.98%
$D = 5$	94.64%
$D = 6$	95.93%
$D = 7$	95.55%

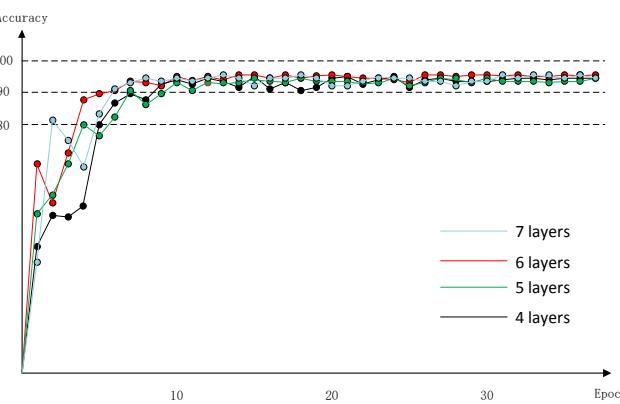


Fig. 6: Comparison of CNN models with different dimensions.

From the above results, it is apparent that $D = 6$ is the ideal choice with the best evaluation accuracy. In addition, it appears

that the size of dimensions has no impact on the convergence speed of the models. In our experiment, for the same number of epochs, a model with lower dimensions consumes less time for computation. Hence, $D = 4$ is also a potential choice with reasonable classification accuracy if the computational speed and efficiency are of concern.

The impact of pooling methods is evaluated in the next experiment. Four CNN models with $D = 6$ are trained with different pooling strategies. Since there are only two pooling methods, the first two models are built with average pooling only and max pooling only. For the remaining two models, one is set with average pooling for the first 3 pooling layers and max pooling for the last 3 pooling layers. The other model is the reverse version of the last model as the control group. The testing accuracy is reported in Table VII and Fig. 7.

TABLE VII: Classification accuracy of different pooling strategies

Pooling strategy	Estimation accuracy
'maximize' for all	95.93%
'average' for all	87.73%
'maximize' for only first 3 pooling	98.29%
'average' for only first 3 pooling	92.5%

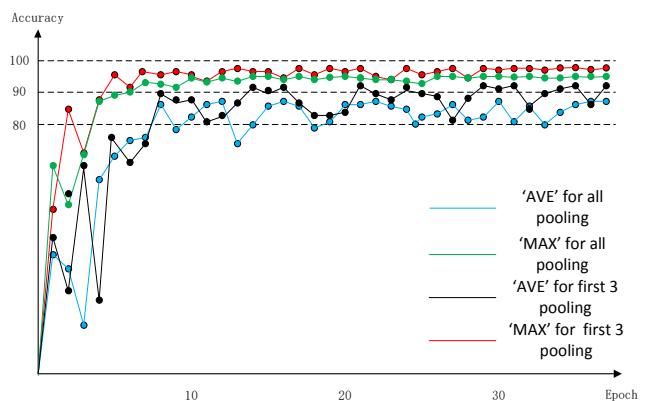


Fig. 7: Comparison of CNN models with different pooling strategies.

In general, if all the pooling layers applied the same pooling method, the max pooling provides more stability for the convergence of the CNN, and the prediction accuracy is also higher if max pooling is selected for all pooling layers. Meanwhile, there appears to be no impact for the convergence speed regardless of the pooling method. In many papers on digital forensics, max pooling is the preferred method to achieve the best performance. Here, we again prove that max pooling is more suitable for solving digital forensics problems. We also noticed that the combined application of different pooling methods provides more advantages if the last 3 pooling layers are set with the 'average' method. The detection is even 2% higher than the model that applied 'maximize' for all pooling layers. After the comparison, in our proposed method, we only set 'maximize' for the first 3 pooling layers and left the other pooling layers with 'average'.

Finally, we evaluated the impact of activation functions. Note that it is a shared perception by many scientists that the 'sigmoid' function is no longer suitable in CNNs but that

'ReLU' is a perfect substitution, and we rescind 'sigmoid' as a choice for activation function. There are also other activation functions that can deliver theoretically better performance than 'ReLU'; however, they are not proven in implementations. Thus, these controversial activation functions are also not considered. Hence, for the activation function, we focus on analyzing the two most applied algorithms: 'ReLU' and 'TanH'. Similar to the above experiment, we also trained four models. The first is activated by 'ReLU', while 'TanH' is selected for all activation layers in the second model. For the mixed activation model, in the third model, we set 'ReLU' for the first 3 activation layers and 'TanH' for the last 3 activation layers. Finally, the fourth model is the reverse version of the third model as the control model. The four models are also trained and tested for the evaluation of 'bicubic' resampled images. The results are displayed in Table VIII and Fig. 8.

TABLE VIII: Classification accuracy of different activation strategies

Activation strategy	Estimation accuracy
'ReLU' for all	98.29%
'TanH' for all	97.57%
'ReLU' for only first 3 activations	96.21%
'TanH' for only first 3 activations	97.07%

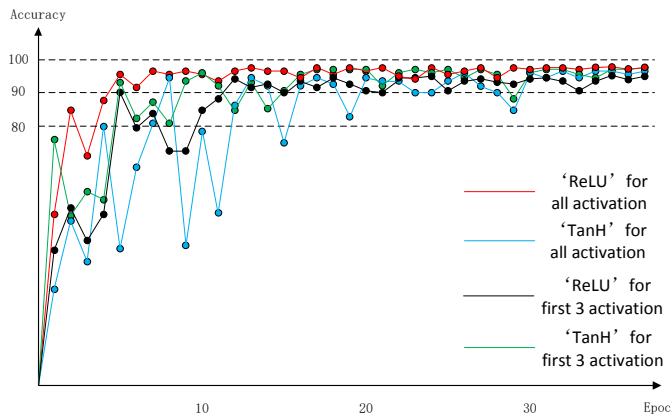


Fig. 8: Comparison of CNN models with different activation strategies.

The results indicate that for all layers, 'ReLU' is the most ideal activation function. Unlike pooling, the strategy for the combined application of different activation functions does not bring any advantages for estimation accuracy. However, the convergence performance is strongly related to the activation function. In general, 'TanH' will introduce oscillations for the convergence, while 'ReLU' can offer stability for the model against this problem. Therefore, all activation layers are finally switched to 'ReLU'.

To summarize the above experiments and discussions, we propose our model with the best overall performance, as shown in Fig. 1. There are also other parameters, such as learning rate, momentum, stride, and kernel size. As we implied above, these parameters more likely belong to the category of fine-tuning, which is beyond our discussion in this paper.

VI. CONCLUSION

In this paper, a method based on a CNN model is proposed to estimate the resampling rate in resampled images. The proposed model is simple in structure; however, it is also highly efficient in estimating the resampling rate. The experiments demonstrate that the proposed method can be employed as a trustworthy resampling rate estimator. The key components, such as pooling strategy, activation functions and the dimensions to build our model, are investigated and determined via experiments and discussions. The most important part of the proposed method, the METEOR layer, is verified to be the key to success for the model. The METEOR layer can enhance the estimation performance in various ways. Improved accuracy, more stable convergence and robustness against JPEG compression are the major contributions of the METEOR layer. All these assessments lead to the conclusion that the proposed method is outstanding for estimating the resampling rate. The problem of parameter estimation via deep learning for other image manipulations is also a potential and inspiring topic for which many further works can be performed compared to the traditional mathematical model analyses.

REFERENCES

- [1] H. Farid, "Digital image forensics," *Scientific American*, vol. 298, no. 6, pp. 66–71, 2008.
- [2] J. Fridrich, "Digital image forensics," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 26–37, 2009.
- [3] S. Lyu and H. Farid, "How realistic is photorealistic?" *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 845–850, 2005.
- [4] A. J. Fridrich, B. D. Soukal, and A. J. Lukáš, "Detection of copy-move forgery in digital images," in *in Proceedings of Digital Forensic Research Workshop*. Citeseer, 2003.
- [5] J. Bunk, J. H. Bappy, T. M. Mohammed, L. Nataraj, A. Flenner, B. Manjunath, S. Chandrasekaran, A. K. Roy-Chowdhury, and L. Peterson, "Detection and localization of image forgeries using resampling features and deep learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2017, pp. 1881–1889.
- [6] S. Prasad and K. Ramakrishnan, "On resampling detection and its application to detect image tampering," in *2006 IEEE International Conference on Multimedia and Expo*. IEEE, 2006, pp. 1325–1328.
- [7] S. Lyu and H. Farid, "Detecting hidden messages using higher-order statistics and support vector machines," in *International Workshop on Information Hiding*. Springer, 2002, pp. 340–354.
- [8] C. Zhang, D. Du, L. Ke, H. Qi, and S. Lyu, "Global contrast enhancement detection via deep multi-path network," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 2815–2820.
- [9] F. Ding, G. Zhu, J. Yang, J. Xie, and Y.-Q. Shi, "Edge perpendicular binary coding for USM sharpening detection," *IEEE Signal Processing Letters*, vol. 22, no. 3, pp. 327–331, 2015.
- [10] F. Ding, G. Zhu, W. Dong, and Y.-Q. Shi, "An efficient weak sharpening detection method for image forensics," *Journal of Visual Communication and Image Representation*, vol. 50, pp. 93–99, 2018.
- [11] S. Li, Z. Han, Y. Chen, B. Fu, C. Lu, and X. Yao, "Resampling forgery detection in JPEG-compressed images," in *2010 3rd International Congress on Image and Signal Processing*, vol. 3. IEEE, 2010, pp. 1166–1170.
- [12] Q. Liu, "An approach to detecting JPEG down-recompression and seam carving forgery under recompression anti-forensics," *Pattern Recognition*, vol. 65, pp. 35–46, 2017.

- [13] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.
- [14] A. Swaminathan, M. Wu, and K. R. Liu, "Digital image forensics via intrinsic fingerprints," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 101–117, 2008.
- [15] G. Cao, Y. Zhao, R. Ni, and A. C. Kot, "Unsharp masking sharpening detection via overshoot artifacts analysis," *IEEE Signal Processing Letters*, vol. 18, no. 10, pp. 603–606, 2011.
- [16] G. Cao, Y. Zhao, R. Ni, and X. Li, "Contrast enhancement-based forensics in digital images," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 3, pp. 515–525, 2014.
- [17] J. Yang, J. Xie, G. Zhu, S. Kwong, and Y.-Q. Shi, "An effective method for detecting double JPEG compression with the same quantization matrix," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 11, pp. 1933–1942, 2014.
- [18] 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.
- [19] M. Boroumand and J. Fridrich, "Deep learning for detecting processing history of images," *Electronic Imaging*, vol. 2018, no. 7, pp. 1–9, 2018.
- [20] T. Bianchi and A. Piva, "Detection of non-aligned double JPEG compression with estimation of primary compression parameters," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 1929–1932.
- [21] T. Pevný and J. Fridrich, "Estimation of primary quantization matrix for steganalysis of double-compressed JPEG images," in *Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, vol. 6819. International Society for Optics and Photonics, 2008, p. 681911.
- [22] N. A. Dodgson, "Image resampling," University of Cambridge, Computer Laboratory, Tech. Rep., 1992.
- [23] H. Cheng, J. Wei, C. Lin, and J. Ye, "Detecting seam-carved image by extreme learning machines using patch analysis method, jury voting, and combinatorial fusion," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.
- [24] D. Zhang, T. Yin, G. Yang, M. Xia, L. Li, and X. Sun, "Detecting image seam carving with low scaling ratio using multi-scale spatial and spectral entropies," *Journal of Visual Communication and Image Representation*, vol. 48, pp. 281–291, 2017.
- [25] J. Wei, Y. Lin, and Y. Wu, "A patch analysis method to detect seam carved images," *Pattern Recognition Letters*, vol. 36, pp. 100–106, 2014.
- [26] T. Yin, G. Yang, L. Li, D. Zhang, and X. Sun, "Detecting seam carving based image resizing using local binary patterns," *Computers & Security*, vol. 55, pp. 130–141, 2015.
- [27] J. Li, X. Li, B. Yang, and X. Sun, "Segmentation-based image copy-move forgery detection scheme," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 507–518, 2015.
- [28] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 758–767, 2005.
- [29] X. Feng, I. J. Cox, and G. Doerr, "Normalized energy density-based forensic detection of resampled images," *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 536–545, 2012.
- [30] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*. ACM, 2016, pp. 5–10.
- [31] A. Piva, "An overview on image forensics," *ISRN Signal Processing*, vol. 2013, 2013.
- [32] M. C. Stamm, M. Wu, and K. R. Liu, "Information forensics: An overview of the first decade," *IEEE Access*, vol. 1, pp. 167–200, 2013.
- [33] Q. Liu and Z. Chen, "Improved approaches with calibrat-
- ed neighboring joint density to steganalysis and seam-carved forgery detection in JPEG images," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 4, p. 63, 2015.
- [34] K. Wattanachote, T. K. Shih, W.-L. Chang, and H.-H. Chang, "Tamper detection of JPEG image due to seam modifications," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2477–2491, 2015.
- [35] M. Kirchner, "Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue," in *Proceedings of the 10th ACM Workshop on Multimedia and Security*. ACM, 2008, pp. 11–20.
- [36] B. Bayar and M. C. Stamm, "Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2691–2706, 2018.
- [37] D. Vázquez-Padín, F. Pérez-González, and P. Comesana-Alfaro, "A random matrix approach to the forensic analysis of upscaled images," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 9, pp. 2115–2130, 2017.
- [38] X. Liu, W. Lu, Q. Zhang, J. Huang, and Y.-Q. Shi, "Downscaling factor estimation on pre-JPEG compressed images," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019. [Online]. Available: <http://dx.doi.org/10.1109/TCSVT.2019.2893353>
- [39] P. Thévenaz, T. Blu, and M. Unser, "Image interpolation and resampling," *Handbook of Medical Imaging, Processing and Analysis*, pp. 393–420, 2000.
- [40] R. C. Gonzalez and R. E. Woods, *Digital image processing (2nd edition)*. Upper Saddle River, NJ: Prentice-Hall, 2002.
- [41] R. E. Carlson and F. N. Fritsch, "Monotone piecewise bicubic interpolation," *SIAM journal on numerical analysis*, vol. 22, no. 2, pp. 386–400, 1985.
- [42] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.
- [43] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1449–1457.
- [44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [45] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.
- [46] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," *arXiv preprint arXiv:1811.12231*, 2018.
- [47] P. Bas, T. Filler, and T. Pevný, "Break our steganographic system: The ins and outs of organizing boss," in *International Workshop on Information Hiding*. Springer, 2011, pp. 59–70.
- [48] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, "Raise: a raw images dataset for digital image forensics," in *Proceedings of the 6th ACM Multimedia Systems Conference*. ACM, 2015, pp. 219–224.



Feng Ding received the B.S. degree from Huazhong University of Science and Technology. He received the M.S. and Ph.D. degrees in Electrical and Computer Engineering from New Jersey Institute of Technology, New Jersey, US. He is currently an Post-doc researcher in State University of New York Albany. His research interests mainly include machine learning, digital forensics and digital image processing.



Hanzhou Wu received both B.Sc. and Ph.D from Southwest Jiaotong University, Chengdu, China, in 2011 and 2017. From 2014 to 2016, he was a visiting scholar in New Jersey Institute of Technology, New Jersey, United States. He was a researcher in Institute of Automation, Chinese Academy of Sciences, from 2017 to 2019. Currently, he is an Assistant Professor in Shanghai University, China. His research interests include information hiding, graph theory and optimization. He has published around 20 papers in peer journals and conferences such as IEEE TDSC, IEEE TCSVT, IEEE WIFS, ACM IH&MMSec, and IS&T Electronic Imaging, Media Watermarking, Security and Forensics.



Guopu Zhu (SM'14) received the B.S. degree in transportation from Jilin University, China, in 2002, and the M.S. and Ph.D. degrees in control science and engineering from the Harbin Institute of Technology, China, in 2004 and 2007, respectively. He was a Postdoctoral Fellow with Sun Yat-sen University, Guangzhou, China, and a Senior Research Associate with the City University of Hong Kong, Hong Kong. Dr. Zhu is currently a Professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China, where he is

the Director of the Center for Internet of Things Computing. His main research areas are multimedia security, image processing, and control theory. He has authored or co-authored more than 30 papers in peer-reviewed international journals. He currently serves as an Associate Editor of the Journal of Information Security and Applications. He is a member of the Youth Innovation Promotion Association, Chinese Academy of Sciences.



Yun Q. Shi (M'88-SM'92-F'05) has joined New Jersey Institute of Technology, US, since 1987. He obtained his M.S. degree from Shanghai Jiao Tong University, China; Ph.D. degree from University of Pittsburgh, US. His research interests include data hiding, forensics, information assurance, visual signal processing and communications. He is an author/coauthor of more than 400 papers, one book, five book chapters, and an editor of 10 books. He holds 30 US patents and has become a Fellow of National Academy of Inventors (NAI) since the end of 2017. He has served as an associate editor of IEEE Transactions on Signal Processing, IEEE Transactions on Information forensics and Security, and IEEE Transactions on Circuits and Systems, an editorial board member of a few journals; also served as the technical program chair of IEEE ICME07, a co-technical chair of IEEE MMSP05 as well as IWDW since 2006, a co-general chair of IEEE MMSP02, a Distinguished Lecturer of IEEE CASS. He has been a member of a few IEEE technical committees, and a Fellow of IEEE since 2005. Since 2018 he has become a life fellow of IEEE.