Image Forgery Detection: An Effective and Low Computational-Cost Data-Driven Model

Thuy Nguyen-Chinh, Thuong Le-Tien, *Member, IEEE,* Hanh Phan-Xuan, Thien Do-Tieu

***Abstract*—Nowadays, as there are a plenty of editing tools that are powerful enough to subtly manipulate images, Image Forensics is more meaningful to detect forged images. Overall, traditional methods tend to solve the problem in a little number of specific assumptions. Therefore, these kinds of methods are not able to cover the whole tampering ways in reality. Besides, solutions based on data-driven have been recently proposed and resulted in prominent results. However, most of such methods are too complex-computational and hungry to data, which requires a large amount of labeled data as well as an energetic hardware for running deep supervised learning.**

In this work, we propose an effective and low computational-cost data-driven model to solve the existing problems. Firstly, a feature extractor using Daubechies Wavelet encodes YCrCb patches in an image as feature vectors of size 450. Subsequently, an artificial neural network classifies these feature vectors whether each one is original or tampered. Specially, we prove that Daubechies Wavelet transform of the luminance (Y) channel is trivial for the classification. Thus, by ignoring luminance features, the computation is reduced by one-third. Experimental results show that the dimensionality reduction leads to a high detection accuracy of 97.11% that is as good as full dimensionality, even better a little bit. Moreover, the method surpasses some conventional solutions and is comparable to deep-learning-based methods even our model suffers in some difficult circumstances (e.g., narrowness, and lack of positive training samples).

***Index Terms*—Forensics, Image Forgery Detection, Artificial Neural Network, Daubechies Wavelet**

I. Introduction

At the present, due to the bloom of digital technology, the amount of multimedia rises significantly, especially images. Nevertheless, along with this growth, there are increasingly powerful tools to manipulate digital images, which may cause critical problems in many cases. Thus, the image forgery detection approach is researched in order to recognize edited images becoming really necessary. Its applications can be seen in legal courts, forensics, social networks, science publications, national intelligence agencies, and so forth.

A. Copy-Move Forgery Detection methods

There is a great deal of proposed methods to solve the problem of Image Forgery Detection, but Copy-Move Forgery Detection is one of the most common approaches because of the typicality in the way creating tampered images. Concretely, a part in an image will be copied and pasted into a different position within the same image. Besides, there may be a post-processing to blur tampering traces. Generally, this approach is divided into two main groups, namely Key-point-based and Block-based.

First, the former [1][2][3] typically extracts features of key points in the image, relying on well-known SIFT and SURF techniques. Then, features of key points are compared in a matching stage to find for similar points. Tampered regions are finally indicated when those ones formed by matching pairs with the same Affine transform over a threshold. This kind of method is helpful in duplication and geometric transform detection because the used techniques as SIFT and SURF own an energetic matching ability to overcome these types of distortion. Nevertheless, in cases of duplicated objects that contains little pattern structure, these two techniques cannot match efficiently, which results in a decline in performance of the detection algorithm.

In the latter method [4][5][6][7], features of blocks, which is generated by sliding window, are extracted from the image. After that, these features are fed into a matching operation, and then blocks are indicated as duplicated regions if matching pairs has a large enough similarity.

Although Copy-Move Forgery Detection is common in Image Forgery Detection, it is only able to handle with cases that the tampered objects are taken inside the same image. This means that it cannot cover cases of splicing where added objects are copied from different sources.

B. JPEG-based methods

Because the most popular image format is JPEG, there is also a huge range of research based on JPEG-format. The key point in this kind of approach is JPEG compression. If someone performs an edition on an original JPEG image, it will occur a double JPEG compression, which is an evidence for scientists to detect traces left on the image.

By exploring the Discrete Cosine transform (DCT), [8] designed a method to detect tampering, based on the DCT double quantization. Its advantages are fast and fine-grained. Moreover, it was the first one automatically localizes tampered regions. Wang *et al.* [9] also used properties of DCT to handle the Image Forgery problem. Under a hypothesis of different distribution of tampered regions, they computed probability of tampered DCT blocks. Besides, they designed 3 types of features to discriminate the true positive samples from the

Thuy Nguyen-Chinh and Thien Do-Tieu are senior students in the Department of Electrical and Electronics Engineering, University of Technology, National University of Ho Chi Minh city, Vietnam (e-mail: {thuy.ng.ch, dotieuthien9997}@gmail.com).

Thuong Le-Tien is with the Department of Electrical and Electronics Engineering, University of Technology, National University of Ho Chi Minh city, Vietnam (e-mail: thuongle@hcmut.edu.vn).

Hanh Phan-Xuan is a Ph.D student in the Department of Electrical and Electronics Engineering, University of Technology, National University of Ho Chi Minh city, Vietnam (e-mail: phantyp@gmail.com).

false positive ones. To detect the recompression in JPEG images, authors in [10] proposed a model to represent the periodic traits in both spatial and DCT domain. This method is able to handle in both of aligned and non-aligned double JPEG compression cases. Thing *et al.* [11] introduced a new periodic detection method of double quantization. Explicitly, they exploited properties of the Gaussian distribution of most significant bins in the DCT histograms of nature images. By exploiting the JPEG ghosts, [12] introduced a method to automatically detect single and double compressed regions. This method is robust in both of aligned and shifted JPEG grids.

In [13], Bianchi *et al.* proposed a Bayesian approach to automatically calculate doubly compressed probability map of 8x8 DCT patches in an image. Because of an assumption that tampered images present a double compression, it requires verifying this assumption before carrying out the detection algorithm. However, unlike previous methods, this work does not need to manually test a suspicious region whether it has a double compression.

Chang *et al.* in [14] proposed a novel algorithm to detect forgery in inpainting images. This method contains two stages, the first one detects suspect regions by searching similar blocks, and then a new method, Multi-Region Relation, is applied to identify tampered regions from output regions of the preceding stage. The strength of this method is fast due to the weight transform, and able to recognize images including uniform background.

In summary, this approach is efficiently solved in the cases of JPEG images and double compression. Nonetheless, in different image formats, it cannot work well because of the employment of recompression in only JPEG images. Therefore, in real situations, this kind of methods may not be applicable.

C. Data-driven methods

Data-driven methods are now increasingly applied in Image Forgery Detection due to the dramatic development of Machine Learning in the last few years. Specifically, this approach feeds a great deal of data into an Artificial Neural Network in order to automatically learn optimal features representing for the data. In [15], instead of extracting features manually, a Convolutional Neural Network (CNN) was firstly used to detect image median filtering forensics. Later, Bayar et al. [16] introduced a new layer in their CNN model to detect manipulation traces. Additionally, Rao et al. [17] proposed a CNN to detect splicing and copy-move forgeries. However, instead of using normal initialization, they assigned a Spatial Rich Model to the first layer to reduce image content while reserving artifacts. Differently, authors in [18] used a transfer-learning approach to point out copy-move forged images by utilizing the AlexNet in [21]. Besides, [19] extracted features of patches within tampered objects by Daubechies Wavelet transform, and then fed them into a Stacked Auto-Encoder so as to classify whether a patch is tampered.

Being inspired of emerging data-driven methods, we propose a data-driven approach to solve the problem of Image Forgery Detection that can clarify cross-contextual situations, which analytical methods cannot address. In particular, a feature extraction method in [19] is utilized, and then an exploration is conducted to analyze the efficiency of the feature extractor. Next, a neural network is to classify these extracted features, accompanying with a method of sample selection in order to help the network learn the distinction between positive and negative samples.

In the following, we will describe our model in section II, and after that, the way to implement our designed model is mentioned in section III. Subsequently, in section IV, experimental results will be discussed before a conclusion in the last section.

II. Proposed method

To solve the problem of Image Forgery Detection, we propose a model with three stages: Feature extraction, Classification, and Post-Processing (Fig. 1).

In the first stage, patches of an image are taken out by a sliding window over the entire an RGB image. Then, these patches are converted into the YCrCb channel. Afterward, the feature extractor in [19], using Daubechies Wavelet transforms, extracts a feature vector of size 450, representing for each patch of the image.

In the second stage, a fully connected neural network is used for classifying whether a patch is tampered. Fig. 2 illustrates the architecture of the proposed neural network. There are total 7 layers, including input, output and 5 hidden layers. The first layer is also the input layer, which has number of neurons corresponding to number of the dimension of a feature vector. Following layers are to encode features from the input layer. Because of nonlinear activations in these layers, nonlinear data can be classified discriminatively, which simple linear models cannot handle. Finally, a softmax layer is added at the end of the neural network to classify the encoded data into two groups (e.g., tampering and non-tampering). Moreover, to tackle with overfitting, dropout [22] is assigned into hidden layers. Totally, this neural network contains 1502 neurons, which is a small number, comparing to other Deep networks. This can reduce the training time as well as boost the testing speed faster.

Lastly, in the third stage, a post-processing is used to obtain a robust conclusion of patches. Concretely, label of a patch will be re-examined by considering surrounding patches. A reliability rate is calculated based on the examined patch and its neighbors. If the reliability rate exceeds a threshold, this patch will be treated as tampering, or non-tampering in otherwise.

III. Implementation

First of all, the neural network is trained to learn how to classify tampered and non-tampered patches, and then this trained network can classify unseen patches. Therefore, in the training process, instead of using sliding window, we reject it and select ourselves content-oriented patches in order to train the neural network. Besides, post-processing is also removed. After training the neural network, the sliding window and the

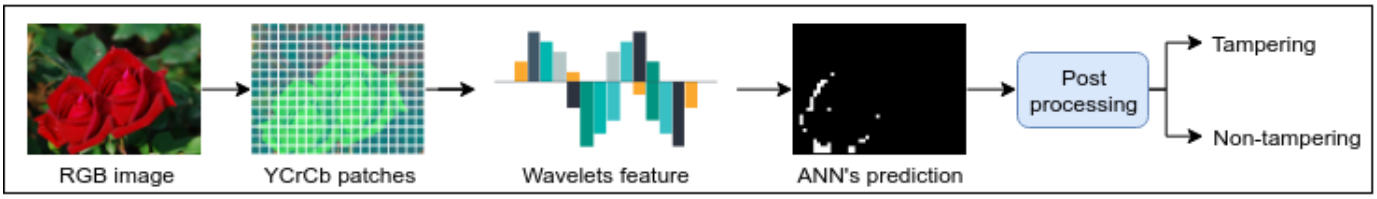


Fig. 1: Flowchart of the proposed method. An RGB image, firstly, is divided into overlapping patches. Then, RGB patches are converted to the YCrCb color channel, before being extracted features using Daubechies Wavelet transforms. Next, a neural network is to classify these patches whether they are forged or not, based on their corresponding feature vectors. Finally, a post-processing stage is designed to fuse a unique conclusion of the examined image.

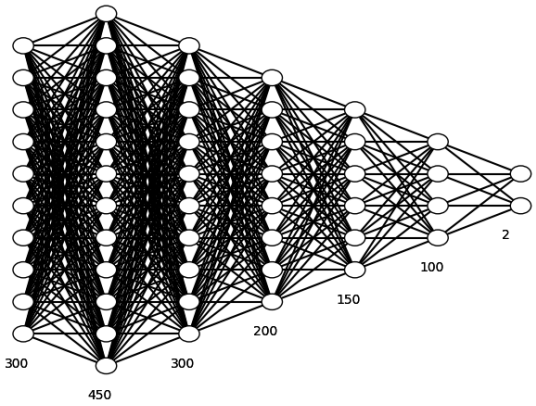


Fig. 2: The proposed neural network for classification. The neural network has 7 layers (including the input and output layers) with totally 1502 neurons.

post-processing will be reused in the testing process. Fig. 3 shows the implementation algorithm log as described.

A. Dataset

To evaluate the performance of our model, we prefer CASIA-v2 database in [20]. This database has one more preceding version. The first version contains 800 authentic and 921 spliced images of a fixed size 384x256 with JPEG format, while the second version consists of 7491 authentic and 5123 tampered images of various sizes from 240x160 to 900x600 with JPEG, BMP, and TIFF formats. According to the authors, comparing to first version, the second one is larger in number of images, diverse in image size, and includes more realistic and challenged fake images.

Initially, two sets of data are prepared (e.g., positive and negative). To assemble the former, with each tampered image, we subtract the tampered to the original one in the YCrCb channel and perform morphological filter on the Y layer to create a Ground Truth (Fig. 4b). Fig. 4a is the tampered images. By subtracting the tampered and original image in the RGB channel, R-, G-, B-channel are obtained in Fig. 4c, 4e, and 4g. Similarly, we also have results in Fig. 4d, 4f, and 4h when conducting in the YCrCb channel. This result shows that the Y channel is quite clear to depict the difference between the original and tampered images, comparing to the others. After that, based on the Ground Truth, patches along boundary of marked regions inside the Ground Truth are selected, which is also the tampering edge.

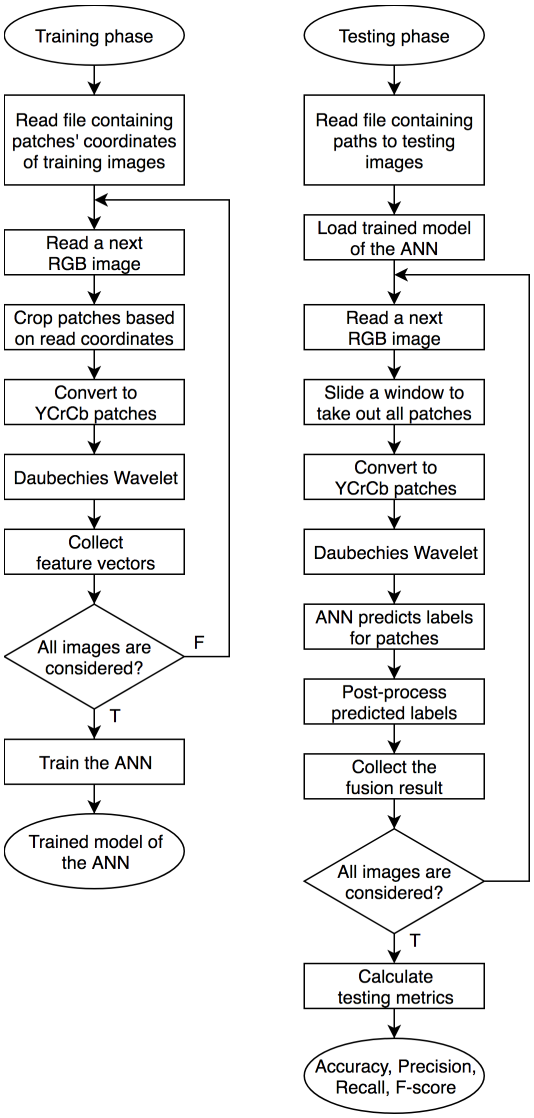


Fig. 3: Algorithm log of the proposed implementation. First, the training process is conducted, outputting a trained model of the neural network. Then, in the testing process, the trained model is used to classify all patches of an image, followed by a post-processing stage to fuse a final conclusion whether an image is forged or not.

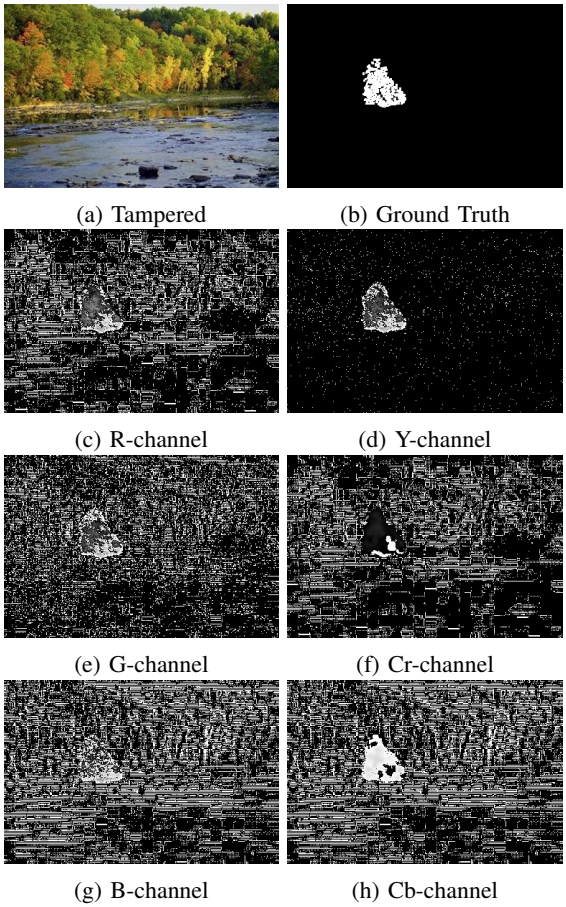


Fig. 4: Compare the efficiency of color channels to create ground truth. From the tampered image in (a) and the corresponding authentic image, a subtraction is performed on color channels, namely R (c), G (e), B (g), Y (d), Cr (f), and Cb (h). As can be seen, the ground truth (b) can be inferred from the result of Y channel (d).

In the point of view, patches lie inside tampered regions may not assist the neural network realize an irregularity because if the forged region is large enough, comparing to the size of the patch, there will be no inconsistency in this patch. Therefore, instead of selecting samples within tampered objects, those ones on the edge of manipulation are chosen. Actually, patches on the tampering boundaries probably contains two different regions. Hence, the neural network can detect this inter-conflict. Furthermore, to build a balanced dataset, while the number of positive samples is quite small, a set of geometric augmentation is applied in order to multiply the amount of positive samples. Nearly 1500 in the overall 5123 tampered images are used to collect positive training data because it is painful to manually select patches on tampering edges. In contrast, it is more simple to create the negative set, i.e. patches inside authentic images are randomly picked. Fig. 5 summaries method of collecting the training set.

B. Feature extraction

Each patch is converted into YCrCb channel, then a filter-bank of 2D Discrete Wavelet Transform is applied to each layer of the YCrCb patch. The Daubechies Wavelet family (db1-db5) is exploited to extract multiscale features of a patch.

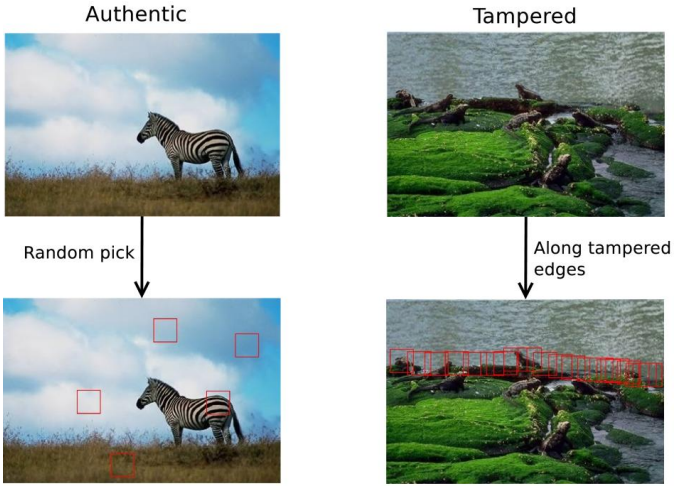


Fig. 5: Method of collecting positive and negative patches. With negative patches, random patches in any position inside the authentic image are automatically selected. Meanwhile, positive patches are carefully labeled along tampering edges inside the tampered image.

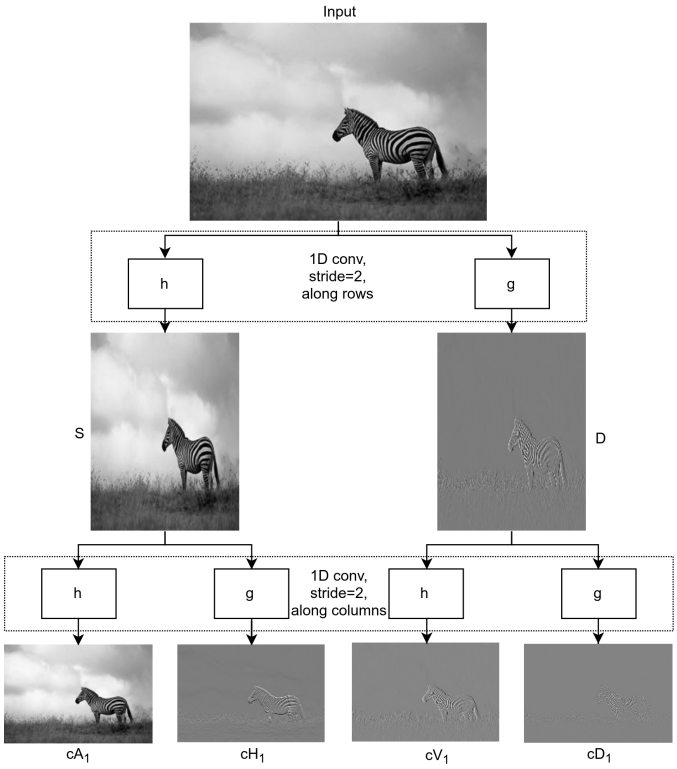


Fig. 6: Procedure of level-1 Wavelet transform. An input is convolved with filters, which results in average and detail component, denoted as S and D. Then, these two latent components are also convolved with the filters to generate four output components, denoted as cA_1, cH_1, cV_1, and cD_1.

Concretely, each transform has two specialized filters, e.g., the low past and high past filter, denoted as h and g, respectively. In which, h and g have a relationship as the following:

$$g[k] = (-1)^k h[M-k] \tag{1}$$

where $k = 1, 2, ..., M$ and M is the length of h and g. Then, these two filters are used to compute 1D convolutions with layers of a patch, along its rows and columns as Fig. 6. Let P is a layer of the patch, S (in Fig. 6) is computed by row-based 1D convolutions:

$$S[i,j] = \sum_{k=1}^{M} h[k]P[i, M+2j-k] \tag{2}$$

Similarly, cA_1 (in Fig. 6) is calculated by column-based 1D convolutions:

$$cA_1[i,j] = \sum_{l=1}^{M} h[l]S[M+2i-l, j] \tag{3}$$

By combining (2) and (3), it becomes:

$$cA_1[i,j] = \sum_{l=1}^{M}\sum_{k=1}^{M} h[l]h[k]P[M+2i-l, M+2j-k] \tag{4}$$

Make a note that the formula to compute cA in (4) is equivalent to a 2D convolution of the patch layer and a 2D filter A with stride of 2:

$$[A]_{M\times M} \triangleq hh^T,\ cA_1 = P * A \tag{5}$$

Therefore, 1D convolutions along rows and columns of the patch layer can be converted to 2D convolutions, which are convenient to image. Similarly, components cH_1, cV_1, and cD_1 (in Fig. 6) also have its corresponding 2D filter:

$$[H]_{M\times M} \triangleq gh^T,\ cH_1 = P * H \tag{6}$$

$$[V]_{M\times M} \triangleq hg^T,\ cV_1 = P * V \tag{7}$$

$$[D]_{M\times M} \triangleq gg^T,\ cD_1 = P * D \tag{8}$$

cA_1, cH_1, cV_1, and cD_1 are called average, horizontal detail, vertical detail, and diagonal detail component of the level-1 wavelet transform. However, among four components, the average cA_1 remains the most information of the input, it is referred as the input of the subsequent step, e.g., the level-2 wavelet transform. Again, aforementioned 2D filters, namely A, H, V, and D are to calculate 2D convolutions with cA_1. This results in level-2 components, namely cA_2, cH_2, cV_2, and cD_2. Lastly, cA_3, cH_3, cV_3, and cD_3 are also computed. One thing desires to make a notice is that size of components in a level is as a quarter as size of components in the previous level, which can capture multiscale features of the image. As a result, we have 10 multiscale components: cH_1, cV_1, cD_1, cH_2, cV_2, cD_2, cA_3, cH_3, cV_3, and cD_3. To condense these 10 matrices, mean (μ), deviation (σ), and sum (Σ) of each matrix are calculated. In total, this work will generate a feature vector of size (3 channels x 5 transforms x 10 result matrices x 3 values), representing for a YCrCb patch.

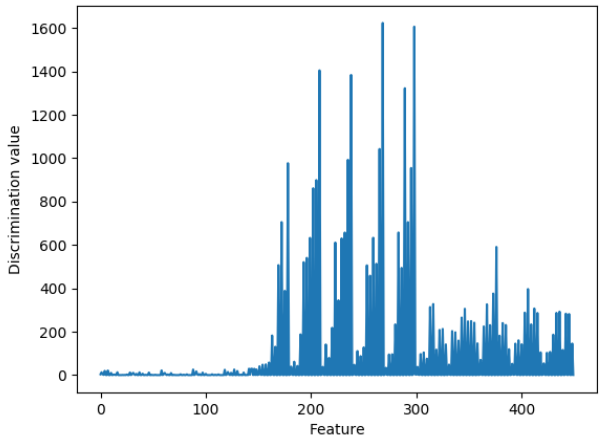


Fig. 7: Discrimination vector represents for the distinction between two set of data, e.g., the positive and negative feature vectors.

To have a clear view about the discrimination trait of the data, we conduct an analysis on extracted feature vectors. Data is normalized, then mean and deviation vectors of two classes of normalized data are computed, denoted as $\mu_1, \mu_2, \sigma_1, \sigma_2$. So, a discrimination vector can be calculated.

$$d = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \tag{9}$$

As expectation that the positive data should be distinguished to the negative data, elements in the discrimination vector are looked forward to being as large as possible.

Fig. 7 depicts result of (9). It can be seen that the first part of 150 elements is insignificant, while the rest is much greater. As a result, in the first one-third elements, data is joint between the positive and negative class. However, data is quite discriminative in the 300 remaining elements. In addition, the insignificant one corresponds to the Y channel. This can be mathematically proved as the following.

Given a 1D signal $x = [x_1, x_2, ..., x_N]$, where $N = 2^J$, the smooth and detail components of Wavelet transform of x are calculated as the following:

$$s[k] = \sum_{i=1}^{M} h[i]x[M+2k-i] \tag{10}$$

$$d[k] = \sum_{i=1}^{M} g[i]x[M+2k-i] \tag{11}$$

where $k \in 1, 2, ..., N/2$, g and h are defined in section III-B. In the case of Daubechies Wavelet, we have $\sum_{i=1}^{M} h[i] = \sqrt{2}$ and $\sum_{i=1}^{M} h^2[i] = 1$, so mean and variance of s depend on mean and variance of x, respectively [30]:

$$\mu(s) = \underset{k}{\mu}\left(\sum_{i=1}^{M} h[i]x[M+2k-i]\right) = \sum_{i=1}^{M} h[i]\underset{k}{\mu}(x) = \sqrt{2}\mu(x) \tag{12}$$

$$\sigma^2(s) = \underset{k}{\sigma^2}\left(\sum_{i=1}^{M} h[i]x[M+2k-i]\right) = \sum_{i=1}^{M} h^2[i]\sigma_k^2(x) = \sigma^2(x) \tag{13}$$

Likewise, because $\sum_{i=1}^{M} g[i] = 0$ and $\sum_{i=1}^{M} g^2[i] = 1$, we derive the relationship between d and x:

$$\mu(d) = \underset{k}{\mu}\left(\sum_{i=1}^{M} g[i]x[M+2k-i]\right) = \sum_{i=1}^{M} g[i]\underset{k}{\mu}(x) = 0 \tag{14}$$

$$\sigma^2(d) = \underset{k}{\sigma^2}\left(\sum_{i=1}^{M} g[i]x[M+2k-i]\right) = \sum_{i=1}^{M} g^2[i]\sigma_k^2(x) = \sigma^2(x) \tag{15}$$

In order to generalize the recent results with 2D signals, we use notations in section III-B. First, the statistic features of cA can be inferred from P:

$$\begin{aligned}\mu(cA) &= \frac{1}{(N/2)^2}\sum_{i=1}^{N/2}\sum_{j=1}^{N/2}[cA]_{ij} = \frac{1}{N/2}\sum_{j=1}^{N/2}\left(\frac{1}{N/2}\sum_{i=1}^{N/2}[cA]_{ij}\right)\\ &= \frac{1}{N/2}\sum_{j=1}^{N/2}\left(\frac{\sqrt{2}}{N}\sum_{i=1}^{N}[S]_{ij}\right) = \frac{\sqrt{2}}{N}\sum_{i=1}^{N}\left(\frac{1}{N/2}\sum_{j=1}^{N/2}[S]_{ij}\right)\\ &= \frac{\sqrt{2}}{N}\sum_{i=1}^{N}\left(\frac{\sqrt{2}}{N}\sum_{j=1}^{N}[P]_{ij}\right) = 2\mu(P)\end{aligned} \tag{16}$$

$$\begin{aligned}\sigma^2(cA) &= \frac{1}{(N/2)^2}\sum_{i=1}^{N/2}\sum_{j=1}^{N/2}\Big([cA]_{ij} - \mu(cA)\Big)^2\\ &= \frac{1}{N/2}\sum_{j=1}^{N/2}\left(\frac{1}{N/2}\sum_{i=1}^{N/2}\Big([cA]_{ij} - \mu(cA)\Big)^2\right)\\ &= \frac{1}{N/2}\sum_{j=1}^{N/2}\left(\frac{1}{N}\sum_{i=1}^{N}\Big([S]_{ij} - \mu(S)\Big)^2\right)\\ &= \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{N/2}\sum_{j=1}^{N/2}\Big([S]_{ij} - \mu(S)\Big)^2\right)\\ &= \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{N}\sum_{j=1}^{N}\Big([P]_{ij} - \mu(P)\Big)^2\right) = \sigma^2(P)\end{aligned} \tag{17}$$

By equivalent expressions, we obtain $\mu(cH) = \mu(cV) = \mu(cD) = 0$ and $\sigma^2(cH) = \sigma^2(cV) = \sigma^2(cD) = \sigma^2(P)$ for five levels of transform. Assume that mean values of both luminance and chroma channels are similar, and variance values of the chroma are much smaller than one of the luminance. This assumption is reasonable because the Y channel contains lots of detail information while the CbCr channels are smoother. Therefore, deviation elements, corresponding to the luminance of feature vectors, are too variant. This increases value of the denominator of (9), so the discriminator vector is tiny in the part of luminance features.

Also, this can be intuitively explained that manipulated objects in an image typically seem to be natural to human vision, so it is difficult to detect these artifacts when there is too much detail in the image. Consequently, the luminance, which has more information of the image than the two chroma channels, is not robust to detect the tampered patches as the chroma. Therefore, by removing Daubechies Wavelet features of the Y channel, the computational cost will be reduced.

C. Classification

The neural network to classify data is sketched in the Fig. 2. First, weights and biases are initialized by Xavier initialization [23] instead of random initialization in order to get a faster training convergence. Also, Xavier initialization ensures that initialized values of weights and biases not tiny or enormous, which may damage the back-propagation during the training process. Moreover, in middle layers, Leaky Rectifier Linear Unit is the activation function [24][25][26] to speed up the computation as well as avoid dead gradient because of the flat left-side edge of the original ReLU activation. After collecting patches from 1500 tampered and 6734 authentic images, a dataset of size 399046 patches is constructed, in which, there are 198520 positive and 200526 negative patches. Make a notice that this database is quite balanced, so the neural network will not tend to be partial to one side. Subsequently, this large dataset is separated into two parts (e.g., training and evaluating set). 90 percent of the whole dataset will be grouped into the training set, subject to portions of positive and negative samples are equal. Then, the rest belongs to evaluating set. The reason for creating a more evaluating set is that the training dataset is just used for training parameters of the neural network. Therefore, in cases of choosing hyper-parameters such as number of epochs, post-processing threshold, dropout value, we must prepare the evaluating dataset to accomplish.

Before training, data is normalized again by computing the mean and standard variance vectors of the training set:

$$x_{mean} = \frac{1}{N_{train}}\sum_{i=1}^{N_{train}} x_i^{(train)} \tag{18}$$

$$x_{dev} = \sqrt{\frac{1}{N_{train}}\sum_{i=1}^{N_{train}}(x_i^{(train)} - x_{mean})^2} \tag{19}$$

Subsequently, the whole training data is normalized:

$$X_{train} = \frac{X_{train} - x_{mean}}{x_{dev}} \tag{20}$$

From here, mean and standard variance vectors in (18) and (19) are stored on the disk as parameters of the neural network. When performing evaluating or testing, these two vectors are loaded so as to normalize the evaluating and testing data. Finally, the training task is done using PyTorch framework on a Quad-Core-i7@2.8GHz laptop, integrated an 8GiB DDR4 RAM and a NVIDIA Geforce 1050 GPU.

D. Post-processing

This last stage is only used for testing examination. First, we manually choose 757 authentic and 800 tampered images, which are not seen by the neural network during the training process, subject to cover almost tampering methods in the

whole types of images. For each image, a sliding window with stride 16 is applied to take out patches of size 32x32. Following that, patches are converted into YCrCb color channel and feature vectors are extracted using Daubechies Wavelet transform. After passing the neural network, a list of labels corresponding to patches appears at the output of the neural network. Then, the Post-processing is utilized to filter out positive labels, which are not reliable, based on information of neighborhood labels.

With a patch, it may have maximum 8 neighbors (patches in corners and border of the image may have less neighbors). Assume that the patch p_0, which owns its label $l(p_0) = 1$, has k neighbors, denoted as $p_i(i = \overline{1,k})$, so the reliability rate can be calculated as the following:

$$Reliability = \frac{1}{k+1}\sum_{i=0}^{k} l(p_i) \tag{21}$$

Subsequently, if the reliability of a patch exceeds a threshold α $(0 \leq \alpha \leq 1)$, its label will remain stable, if not, the label will change to negative.

Lastly, a simple fusion operation is to decide whether an image is tampered. If total patches within the image are negative, the image will be negative to forgery. In contrast, if there is at least one tampered patch, the image is indicated as forgery.

IV. Experimental results

In this section, we define some metrics for evaluating the model. The result of classification will be in 4 possible cases, namely True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Some formulas below are metrics that we will use. All of them are in the range of [0,1].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

$$Precision = \frac{TP}{TP + FP} \tag{23}$$

$$Recall = \frac{TP}{TP + FN} \tag{24}$$

$$Fscore = \frac{2 * Precision * Recall}{Precision + Recall} \tag{25}$$

In these four metrics, the $Accuracy$ represents a general information of the models performance. However, in anomaly detection problems, the number of positive samples are typically greatly smaller than the negative ones. Consequently, if the model is simply set in a way that all of inputs are classified as negatives, it can reach a spectacular accuracy. Therefore, $Precision$ and $Recall$ are exploited to overcome the shortcoming of $Accuracy$. To be more clear, $Precision$ reflects how many samples that are exactly positive among samples indicated as positive, whilst $Recall$ highlights the ratio of samples predicted as positive inside definite positive samples. Besides, we hope that there is a unique metric, representing ability of a model in the problem of skewed distribution detection, instead of two these metrics of $Precision$ and $Recall$. Fortunately, $Fscore$ is an answering one. In its formula, we can see that $Fscore$ contains information of both $Precision$ and $Recall$. Besides, the range of $Fscore$ is from 0 to 1, which is normalized to be relevant to probability. All of these four metrics are expected as asymptotic to one as possible.

TABLE I. METRICS OF THE TRAINING AND TESTING PROCESS

Metric	Training	Testing
Accuracy	98.21%	97.11%
Precision	99.08%	98.88%
Recall	97.32%	95.65%
F-score	98.19%	97.23%

A. Training process

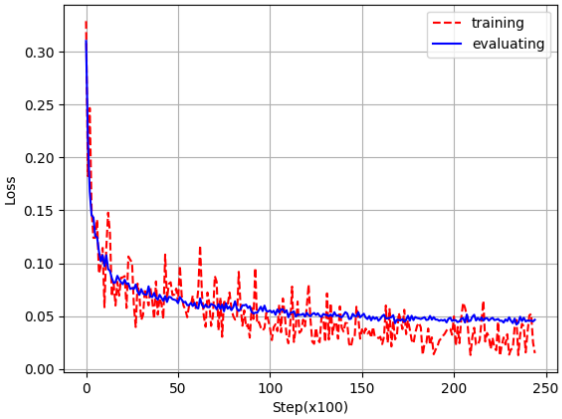
In the training process, the main target is to train the neural network so that it can learn optimal parameters itself in order to classify data into two classes. In this task, the Adam optimizer [27] is used with the learning rate $1e^{-3}$, and epoch decay factor of 0.95. After 35 epochs, the result is shown in Fig. 8. As can be seen, the training lines are not stable, fluctuating around the evaluating lines. This is caught by the dropout followed layers in the neural network. In the time of training, within each iteration, some of neurons in a layer will be randomly chosen to be deactivated. Moreover, their weights and biases are also not updated by back-propagation. This will lead to a fair training that no neuron is too active while the others are inactive. As a result, the training loss and accuracy will fluctuate because of the deactivation of random neurons. Nevertheless, with evaluating set, the dropout is not used, so the evaluating lines are stable.

Table I reveals results of the last step (these metrics are computed on the evaluating set, not the training set, and the computational unit is patch). All of metrics are equally high, which reflects the robustness of our neural network. This training process frequently took us less than 5 minutes to train.

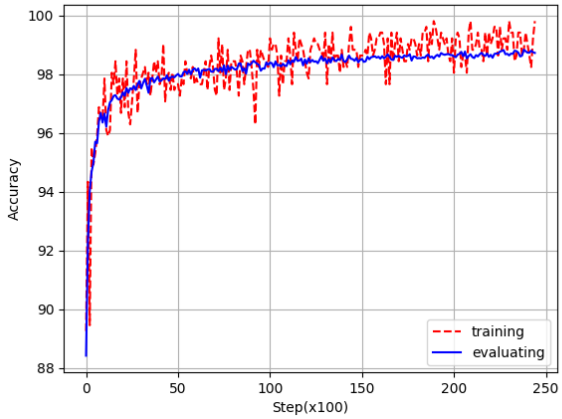
B. Testing process

Testing process is conducted after training the neural network. There are 757 negative and 800 positive images used in this process. First, by sliding a window, overlapping patches are taken out. Then, patches in YCrCb are transformed in the Wavelet domain and fed into the neural network. After the neural network predicts labels for patches, a post-processing is applied in order to conclude whether the image is tampered. Finally, results are shown in Table I. These metrics are computed on the unit of image that is different from the training process, where computational unit is patch. The reason for this difference is that in the training process, we manually pick content-oriented patches to train the neural network. By contrast, in the testing process, post-processing and fusion are added to decide a final conclusion of images, so results represent for images, not patches.

Although the final result of our model is detecting whether an image is forged or not, we also visualize binary maps of classified patches. Fig. 9 draws testing results of some images. Here, there are totally four columns (e.g., origin,

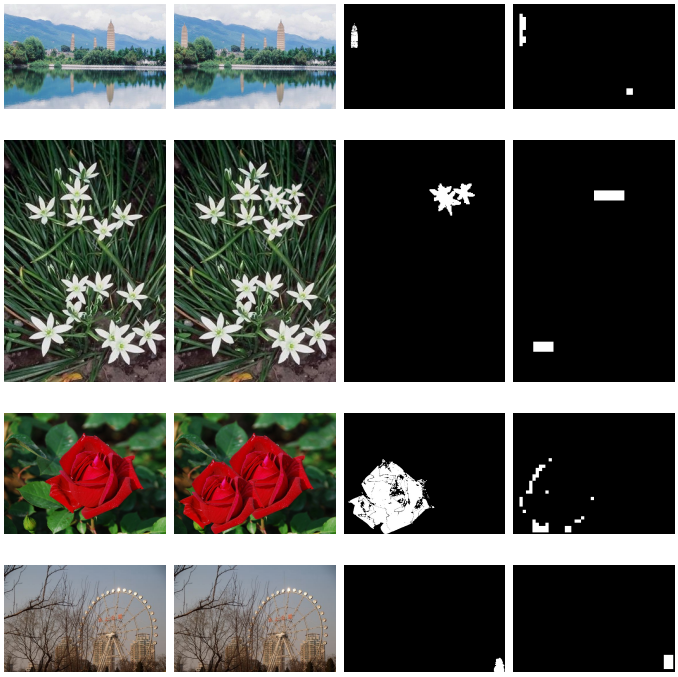


(a) Loss



(b) Accuracy

Fig. 8: Loss and accuracy during the training process. These metrics are calculated on both of training set and evaluating set.



(a) Origin (b) Tampering (c) Gnd Truth (d) Prediction

Fig. 9: Testing predictions of some images. Tampered images, ground truths, and predictions are depicted in the first, second, and third columns, in turn.

TABLE II. COMPARISON BETWEEN TWO NEURAL NETWORK MODELS

Metrics	**Training process**		**Testing process**	
	input-300	*input-450*	*input-300*	*input-450*
Accuracy	98.21%	98.68%	97.11%	96.92%
Precision	99.08%	99.00%	99.75%	99.38%
Recall	97.32%	98.31%	94.89%	94.87%
F-score	98.19%	98.65%	97.26%	97.07%
Time	$4m0s$	$5m5s$	$3.570s$	$3.757s$

tampering, ground truth, and prediction), each one contains four images. The predictions are quite well matched to ground truths. Explicitly, our proposed neural network is trained by positive samples, which are patches on edges of tampering operation, so predictions will mark positions on the tampering boundaries. For instance, in the third row, a new rose is added into the origin, which is easily realized by seeing the ground truth. Because of the way of training dataset selection, the prediction points out a tampering edge around the pasted flower. Besides that, the neural network is also able to recognize small objects. The last row demonstrates this ability of our neural network. There is a tiny object on the corner of the image, and our model can detect it in the prediction.

C. Evaluate the efficiency of the dimensionality reduction

Besides the neural network accomplishing with 300-D feature vectors in Fig. 2, we also run an experiment on a different neural network (Fig. 10), which is same as the original network, excepting the input layer consists of 450 neurons. We denote the neural network in Fig. 2 as input-300 model and the new neural network as input-450 model. Purpose of this work is proving that 300-D feature vectors, which are dimensionally reduced, are as effective as original 450-D vectors. These two neural networks are trained in the same configuration, namely training dataset, and number of epoch. Besides, they are also tested under a same testing dataset, including 757 authentic and 800 tampered images.

Fig. 11 plots two evaluating accuracy lines versus epoch during the training process. Obviously, the input-300 model has a sharp approximate to the input-450 model, but slightly under. Additionally, in Table II, metric values are recorded on both of training and testing process. Those testing ones in the first model are better than the second ones. This outperformance can be explained that the first model is able to learn generalizable features because its training dataset is selected to be distinguished. As a result, while the input-450 reach a higher accuracy in the training process, the input-300 model, however, has all higher metrics in the testing process. Hence, by reducing unnecessary features, the model is still able to remain the final performance, while the computational speed is improved.

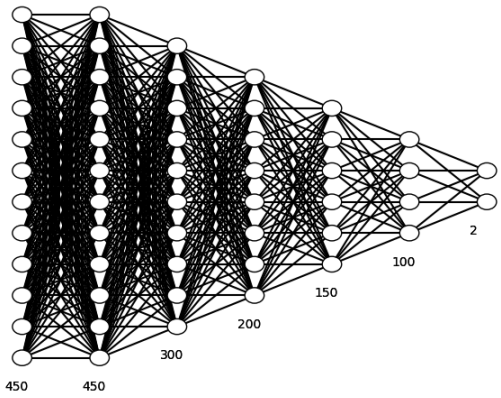


Fig. 10: A different neural network. This neural network is same as the original neural network in Fig. 2, excepting the number of neuron in the first layer. While the original one has 300 neurons, this neuron network has 450 neurons inputted to the first layer.

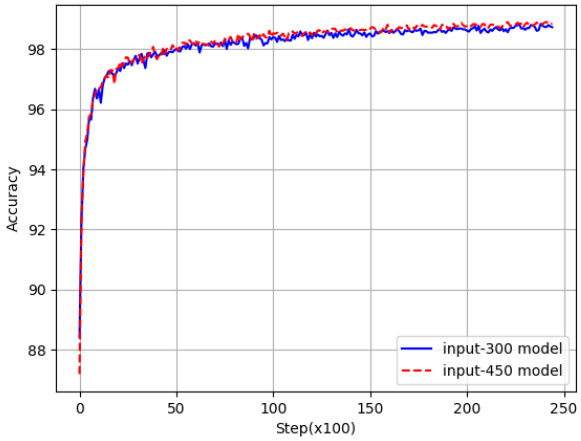


Fig. 11: Accuracies of two models during the training process. These metrics are calculated on the evaluating set.

D. Compare to different methods

After testing the proposed model, we continue comparing our performance to the others. Two conventional methods [28][29] are chosen, alongside with one more data-driven method [17], to make a detection comparison. This comparison uses accuracy metric obtained when testing on the CASIA-v2 database. In Table III, our method stands at the second rank, which overcomes two conventional ones and left behind the data-driven one. Concretely, methods of [17], mine, and [28] are quite approximate, while the last one in [29] is far from the top results. This comparison reveals that two data-driven methods outperform those ones of convention.

As regards the two data-driven methods, in [17], Rao *et al.* used a powerful CNN model with 10 layers and a SVM classifier at the end of pipeline as well as they utilized the whole CASIA-v2 database for training and testing. Meanwhile, due to the pain of the manually sample selection, we can merely prepare 1000 tampered images to train that is much smaller than the training set of [17]. In addition, our model is also too narrow, comparing to the one of [17]. The model of Rao et al. took about 1 hour for training on the NVIDIA Tesla K40 GPU, whereas, our model just requires around 4 minutes for training on the NVIDIA Geforce 1050 GPU. However, the two accuracies are fairly equal. This demonstrates that our model can suffer the data hunger as being seen among Deep networks. Besides, because of the narrowness in the architecture, our proposed model is probably faster than other Deep networks in both of training and testing process, while it can keep a high accuracy.

TABLE III. DETECTION COMPARISON BETWEEN METHODS ON THE CASIA-V2 DATABASE

Method	Accuracy	Number of test images	Number of neurons
Rao et al. [17]	97.83%	2102	606752
Proposed	**97.11%**	**1557**	**1502**
Goh et al. [28]	96.21%	1200	-
He et al. [29]	87.37%	1200	-

V. Conclusion

In this paper, we proposed a low computational-cost and effective data-driven model to solve the problem of Image Forgery Detection. In which, a fully connected neural network, along with relating components (e.g., activations, initialization, normalization, optimizer), was designed to classify tampered patches inside an image. By conducting a discrimination analysis on extracted features, we pointed out that the Daubechies Wavelet features of the luminance channel in YCrCb is less useful for the neural network to classify tampered patches. Therefore, by removing them, the computational cost will be significantly reduced in both of training and testing process. Also, we conducted two experiments to verify the efficiency of our dimensional reduction proposal. In the first one, we obtained the result that the neural network, which learns 300-D features, can perform better in accuracy and time than the neural network that learns 450-D features. This result proves our dimensionality reduction method is relevant. Besides, in the second experiment, we compare our model to some baselines, including two conventional methods and a data-driven method. Our model can achieve a noticeable detection accuracy of 97.11%, while it suffers tough conditions, namely narrowness in architecture and lack of positive data.

In the future, we will explore some features that are more discriminative between tampering and non-tampering, and continue applying dimensionality reduction methods to boost the computational speed. Also, other Deep Learning types, such as CNN and LSTM, will be considered to enhance the classification performance.

References

[1] X. Pan and S. Lyu, *"Region duplication detection using image feature matching"*, IEEE Transactions on Information Forensics and Security, vol. 5, no.4, ISSN: 1556-6013, pp. 857-867, 2010.

[2] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo and G.Serra, *"A sift-based forensic method for copymove attack detection and transformation recovery"*, IEEE Transactions on Information Forensics and Security, vol. 6, no. 3, ISSN: 1556-6013, pp. 1099-1110, 2011.

[3] P. Kakar, N. Sudha, *"Exposing postprocessed copy-paste forgeries through transform-invariant feature"*, IEEE Transactions on Information Forensics and Security, vol. 7, no. 3, ISSN: 1556-6013, pp. 1018-1028, June 2012.

[4] S.-J. Ryu, M.-J. Lee and H.-K. Lee, *"Detection of copy-rotate-move forgery using Zernike moments"*, Information Hiding Conference, Lecture Notes in Computer Science, vol. 6387, Springer, Heidelberg-Berlin, 2010, ISBN: 978-3-642-16434-7.

[5] H.-J. Lin, C.-W. Wang and Y.-T. Kao, *"Fast copy-move forgery detection"*, WSEAS Transactions on Signal Processing, vol. 5, no. 5, ISSN: 0031-3203, pp. 188-1975, 2009.

[6] V. Christlein, C. Riess, J. Jordan and E. Angelopoulou, *"An evaluation of popular copy-move forgery detection approaches"*, IEEE Transactions on Information Forensics and Security, vol. 7, no. 6, ISSN: 1556-6013, pp. 1841-1854, 2012.

[7] T. L.-Tien, T. H.-Kha, L. P.-C.-Hoan, A. T.-Hong, N. Dey, M. Luong, *"Combined Zernike Moment and Multiscale Analysis for Tamper Detection in Digital Images"*, Informatica (An International Journal of Computing and Informatics), vol.41, no.1, ISSN: 0350-5596, March 2017.

[8] Z. Lin, J. He, X. Tang, K. Tang, *"Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis"*, Pattern Recognition, vol. 42, no. 11, ISSN: 0031-3203, pp. 2492-2501, January 2009.

[9] W. Wang, J. Dong, T. Tan, *"Exploring DCT coefficient quantization effects for local tampering detection"*, IEEE Transactions on Information Forensics and Security, vol. 9, no. 10, ISSN: 1556-6013, pp. 1653-1666, October 2014.

[10] L. Chen, T. Hsu, *"Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection"*, IEEE Transactions on Information Forensics and Security, vol. 6, no. 2, ISSN: 1556-6013, pp. 396-406, June 2011.

[11] L. Thing, Y. Chen, C. Cheh, *"An improved double compression detection method for JPEG image forensics"*, In IEEE International Symposium on Multimedia, pages 290-297, December 2012, ISBN: 978-1-4673-4370-1.

[12] F. Zach, C. Riess, and E. Angelopoulou, *"Automated image forgery detection through classification of JPEG ghosts"*, Pattern Recognition, 7476, pp. 185-194, January 2012.

[13] T. Bianchi, A. Piva, *"Image forgery localization via block-grained analysis of JPEG artifacts"*, IEEE Transactions on Information Forensics and Security, vol. 7, no. 3, ISSN: 1556-6013, pp. 1003-1017, June 2012.

[14] C. Chang, C. Yu, C. Chang, *"A forgery detection algorithm for exemplar-based inpainting images using multi-region relation"*, Journal Image and Vision Computing, vol. 31, no. 1, ISSN: 0262-8856, pp. 57-71, MA-USA, 2013.

[15] J. Chen, X. Kang, Y. Liu and Z. J. Wang, *"Median Filtering Forensics Based on Convolutional Neural Networks"*, IEEE Signal Processing Letters, vol. 22, no. 11, ISSN: 1070-9908, pp. 1849-1853, November 2015.

[16] B. Bayar, M. C. Stamm, *"A Deep Learning Approach to Universal Image Manipulation Detection Using a New Convolutional Layer"*, Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, pp. 5-10, New York-USA, 2016, ISBN: 978-1-4503-4290-2.

[17] Rao Yuan, Ni Jiangqun, *"A deep learning approach to detection of splicing and copy-move forgeries in images"*, IEEE International Workshop on Information Forensics and Security (WIFS), Abu Dhabi-United Arab Emirates, 2016, ISBN: 978-1-5090-1139-1.

[18] J.Ouyang, Y.Liu, M.Liao, *"Copy-Move Forgery Detection Based on Deep Learning"*, 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, Shanghai-China, 2017, ISBN: 978-1-5386-1938-4.

[19] Y. Zhang, J. Goh, L. Win, V. Thing, *"Image Region Forgery Detection: A Deep Learning Approach"*, Proceedings of the Singapore Cyber-Security Conference, Singapore, 2016, ISBN: 978-1-61499-616-3.

[20] J. Dong and W. Wang, *"Casia tampering detection dataset"*, 2011.

[21] A. Krizhevsky, I. Sutskever, G. Hinton, *"Imagenet classification with deep convolutional neural networks"*, NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems, vol. 1, pp. 1097-1105, Nevada-USA, 2012, DOI: 10.1145/3065386.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, *"Dropout: a simple way to prevent neural networks from overfitting"*, The Journal of Machine Learning Research, vol. 15, no. 1, ISSN 1533-7928, pp. 1929-1958, January 2014.

[23] Xavier Glorot, Yoshua Bengio, *"Understanding the difficulty of training deep feedforward neural networks"*, Proceedings of the 13rd International Conference on Artificial Intelligence and Statistics, PMLR 9, pp. 249-256, Sardinia-Italy, http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf, 2010.

[24] V. Nair, E. Hinton, *"Rectified Linear Units Improve Restricted Boltzmann Machines"*, Proceedings of the 27th International Conference on Machine Learning, pp. 807-814, Haifa-Israel, 2010, ISBN: 978-1-60558-907-7.

[25] B. Xu, N. Wang, T. Chen, M. Li, *"Empirical Evaluation of Rectified Activations in Convolutional Network"*, https://arxiv.org/abs/1505.00853v2, 2015.

[26] K. He, X. Zhang, S. Ren, J. Sun *"Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification"*, https://arxiv.org/abs/1502.01852v1, 2015.

[27] P. Kingma, J. Ba, *"Adam: A Method for Stochastic Optimization"*, 3rd International Conference for Learning Representations, San Diego-USA, https://arxiv.org/abs/1412.6980, 2015.

[28] J. Goh and V. L. L. Thing, *"A hybrid evolutionary algorithm for feature and ensemble selection in image tampering detection"*, International Journal of Electronic Security and Digital Forensics, vol. 7, no. 1, ISSN: 1751-911X, pp. 76-104, March 2015.

[29] Z. He, W. Lu, W. Sun, J. Huang, *"Digital image splicing detection based on Markov features in DCT and DWT domain"*, Pattern Recognition, vol. 45, no. 12, ISSN: 0031-3203, pp. 4292-4299, 2012.

[30] A. Cohen, T. Tiplica, and A. Kobi, *"Design of experiments and statistical process control using wavelets analysis"*, Control Engineering Practice, vol. 49, ISSN: 0967-0661, pp. 129-183, April 2016.

Thuy Nguyen-Chinh was born in Dong Nai, Vietnam. At the present, he is a senior undergraduate student in the Honor class of Electronics and Telecommunications, Electrical and Electronics Engineering Department, HoChiMinh city University of Technology (HCMUT), Vietnam. His research interests relate to apply machine learning and deep learning techniques to solve problems in signal processing, computer vision, and embedded system, particularly in image forensics, biometric recognition, and autonomous robotics.

Thuong Le-Tien (MIEEE-96) was born in Saigon, HoChiMinh City, Vietnam. He received the Bachelor and Master Degrees in Electronics-Engineering from HoChiMinh City Uni. of Technology (HCMUT), Vietnam, then the Ph.D. in Telecommunications from the Uni. of Tasmania, Australia. Since May 1981 he has been with the EEEng. Department at the HCMUT. He spent 3 years in the Federal Republic of Germany as a visiting scholar at the Ruhr Uni. from 1989-1992. He served as Deputy Department Head for many years and had been the Telecommunications Department Head from 1998 until 2002. He had also appointed for the second position as the Director of Center for Overseas Studies since 1998 up to May 2010. His areas of specialization include: Communication Systems, Signal Processing and Electronic Circuits. He has published more than 150 research articles and the teaching materials for university students related to Electronic Circuits 1 and 2, Digital Signal Processing and Wavelets, Antenna and Wave Propagation, Communication Systems. Currently he is a full professor at the HCMUT.

Hanh Phan-Xuan got the B.Eng and M.Eng. Degrees from the HoChiMinh City University of Technology (HCMUT), Vietnam. His research relates to Image Signal Processing, Neural Networks and Deep Learning Techniques to solve problems in Computer Vision, Image Forgery Detection, Biometrics Signal Processing, and Autonomous Robotics. Currently, he is a Ph.D. Student at EEE Department of the HCMUT.

Thien Do-Tieu is a senior student in the Honor class of Electronics and Telecommunications, Electrical and Electronics Engineering Department, HoChiMinh city University of Technology (HCMUT), Vietnam. He does researches on Machine Learning and Deep Learning in Computer Vision, especially Image Forensics, Face recognition and Autonomous cars.