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

Thuong Le-Tien, Hanh Phan-Xuan, Thuy Nguyen-Chinh, and Thien Do-Tieu

***Abstract*—Nowadays, Image Forgery Detection contributes an indispensable role in digital forensics, while there are increasingly more sophisticated forgery methods. In overall, almost conventional methods just focus on identifying specific features in tampered images, therefore, such methods cannot cover whole possible cases in reality. Recently, some data-driven proposals have been exploited to handle these barriers and attained prominent results. However, almost these ones are hungry to data because of the complication in deep architectures, which requires a large amount of data and an energetic implementation hardware. In this paper, we propose a low computational-cost and effective data-driven model as a modified deep learning-based model to solve the existing problems above. The process of approach is overviewed as follows: Firstly, the Daubechies Wavelet transform is utilized to extract features of size 450, representing YCrCb patches inside the image. Then, a neural network is used to classify forged patches. However, when conducting a discrimination analysis, we found that the luminance channel (Y) does not play an essential role in the forgery detection, whereas, it is better by using two chrominance channels (Cr and Cb). The idea is stated by removing these luminance features, then the feature vector dimension changes to as two-thirds as its origin, which reduces efficiently the computational cost in both of training and testing processes. The experimental results reveal that our proposed method reaches a high detection accuracy of 97.11%, even the model suffers in some difficult circumstances (e.g., narrowness, and lack of positive training samples). As a result, the proposed model is effective to address the mentioned challenges.**

***Index Terms*—Forensics, image forgery detection, neural network, modified deep learning, Daubechies Wavelets.**

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, authorization, etc.

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]-[3] typically extracts features of key points in the image, relying on well-known as Scale-Invariant Feature Transform (SIFT) and Speeded Up Robust Feature (SURF) technique. 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 methods [4]-[7], features of blocks, which is generated by sliding windows, 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 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

Manuscript received July 20, 2018; revised March 8, 2019. This work was supported in part by the HoChiMinh City University of Technology, HCMUT, VietNam under Grant Contract 135/HD-DHBK-KHCN-DA, code TNCS-DDT-2017-27.

The authors are with the Department of Electrical and Electronics Engineering, University of Technology, National University of Ho Chi Minh city, Vietnam (Corresponding Author: Thuong Le-Tien, Hanh Phan-Xuan; e-mail: thuongle@hcmut.edu.vn, phantyp@gmail.com, thuy.ng.ch@gmail.com, dotieuthien9997@gmail.com).

doi: 10.18178/ijmlc.2019.9.2.784

both of aligned and non-aligned double JPEG compression cases. L. 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 8×8 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 in-painting 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. 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.

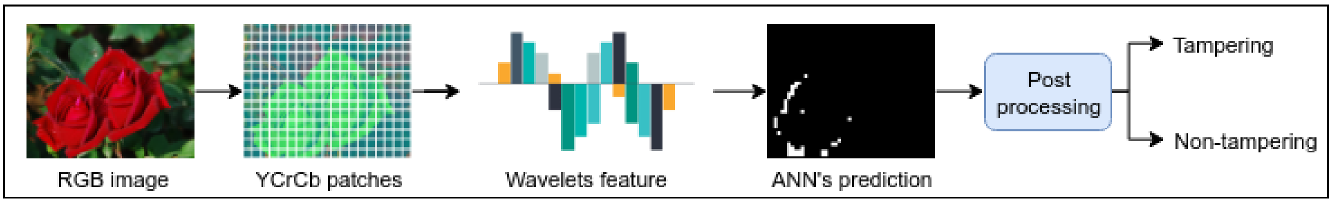


Fig. 1. Flowchart of the proposed method. An RGB image, firstly, is divided into overlapping patches. Then, RGB patches are converted to the YCrCb color channel, before being extracted features using Daubechies Wavelet transforms. Next, a neural network is to classify these patches whether they are forged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

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 [19]. Besides, [20] extracted features of patches within tampered objects by the 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 [20] 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 content, 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 RGB patches are converted into the YCrCb channel. Afterward, the feature extractor, using the Daubechies Wavelet transforms, extracts a feature vector of size 300, representing for each patch of the image.

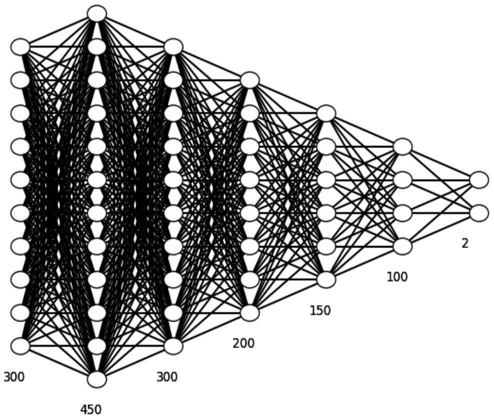


Fig. 2. The proposed neural network for classification. It has 7 layers (including the input and output layers) with totally 1502 neurons.

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 [21] is assigned into hidden layers. 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.

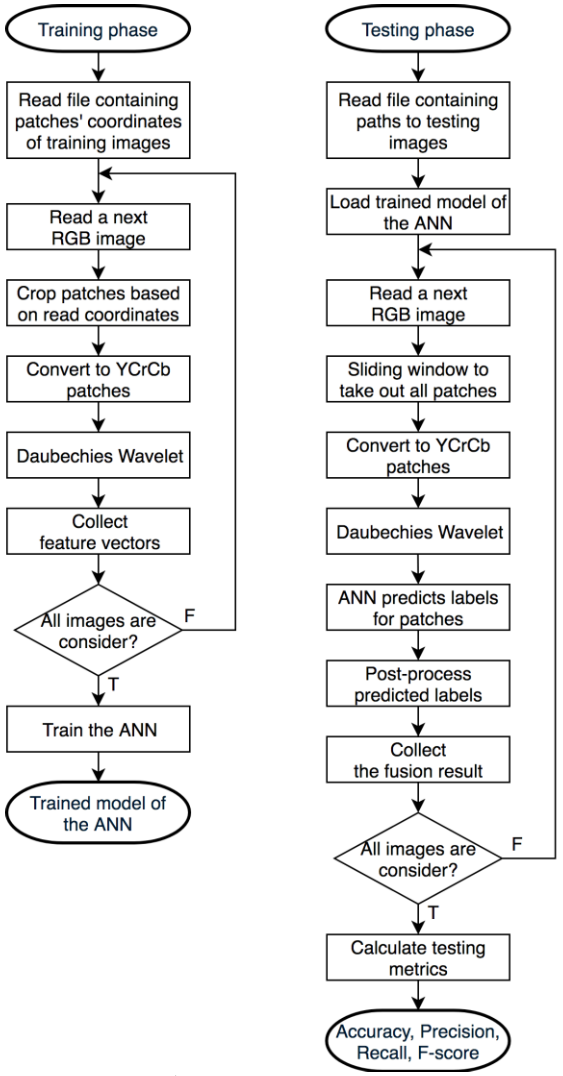

Training phase
Read file containing patches' coordinates of training images
Read a next RGB image
Crop patches based on read coordinates
Convert to YCrCb patches
Daubechies Wavelet
Collect feature vectors
All images are consider?
F
T
Train the ANN
Trained model of the ANN
Testing phase
Read file containing paths to testing images
Load trained model of the ANN
Read a next RGB image
Sliding window to take out all patches
Convert to YCrCb patches
Daubechies Wavelet
ANN predicts labels for patches
Post-process predicted labels
Collect the fusion result
All images are consider?
F
T
Calculate testing metrics
Accuracy, Precision, Recall, F-score

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.

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 itand 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 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 [22]. This database has one more preceding version. The first version contains 800 authentic and 921 spliced images of a fixed size 384×256 with JPEG format, while the second version consists of 7491 authentic and 5123 tampered images of various sizes from 240×160 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.

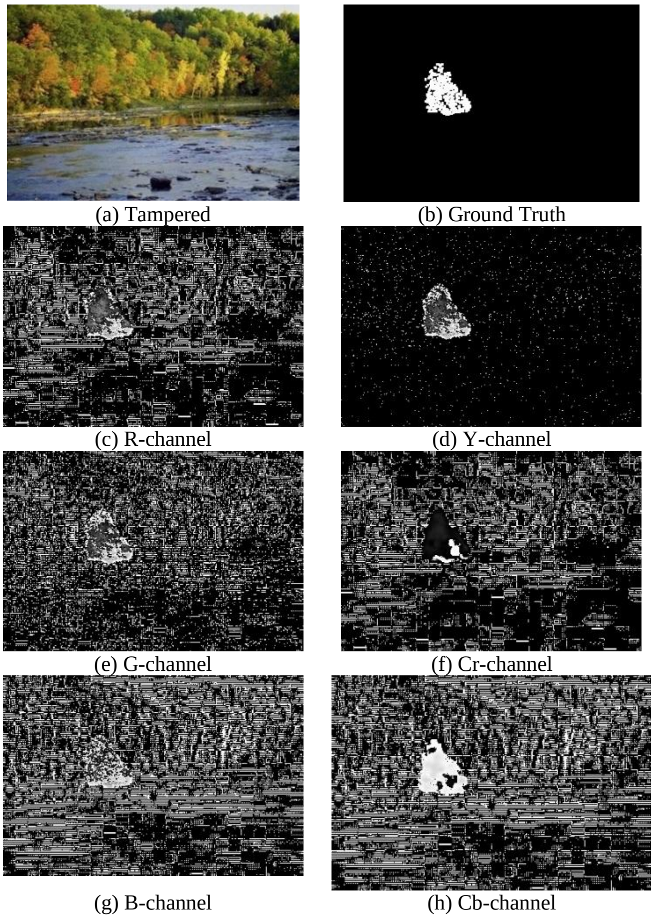

(a) Tampered
(b) Ground Truth
(c) R-channel
(d) Y-channel
(e) G-channel
(f) Cr-channel
(g) B-channel
(h) Cb-channel

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).

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. 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.

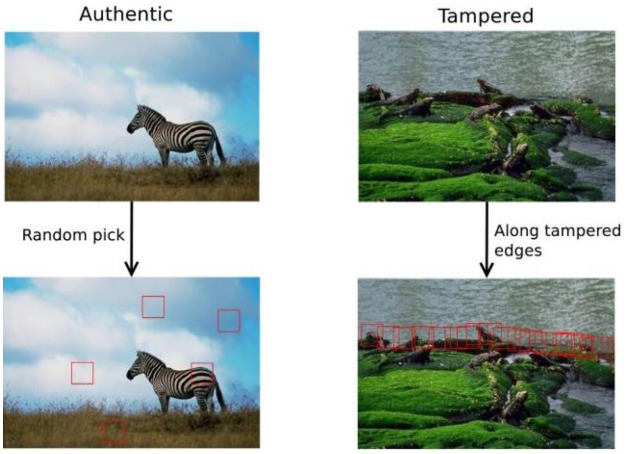


Fig. 5. Method of collecting positive and negative patches. With negative patches, random patches in any position inside the authentic image are automatically selected. Meanwhile, positive patches are carefully labeled along tampering edges inside the tampered image.

B. Feature Extraction

Each RGB patch, which is taken using the mentioned method, is converted into YCrCb channel, then 5 level-3 of the Daubechies Wavelet transforms (db1-db5) are applied to each layer of the YCrCb patch. This work generates 150 matrices (3 channels × 5 transforms × 10 result matrices) for each YCrCb patch. In each matrix, its mean, standard variance, and sum are calculated. These computed values are elements in the feature vector. Consequently, the feature vector, which represents the patch, has its size of 450.

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.

$$d = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \tag{1}$$

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. 6 depicts result of (1). 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. Actually, this can be 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.

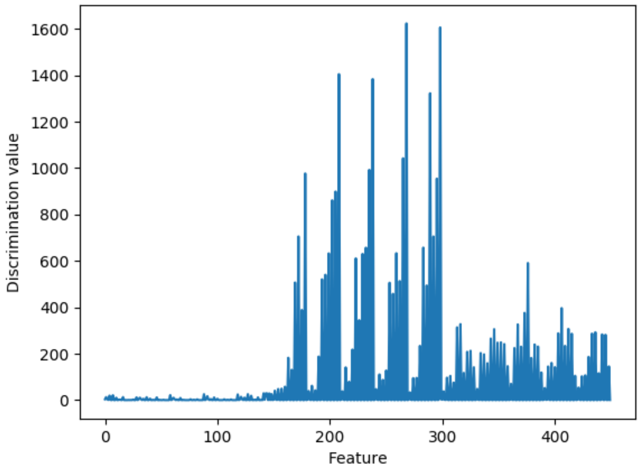


Fig. 6. Discrimination vector represents for the distinction between two set of data, e.g., the positive and negative feature vectors.

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]-[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:

$$x_{\text{mean}} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} x_i^{(\text{train})} \quad (2)$$

$$x_{var} = \sqrt{\frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \left(x_i^{(\text{train})} - x_{\text{mean}}\right)^2} \quad (3)$$

Subsequently, the whole training data is normalized:

$$X_{\text{train}} = \frac{x_{\text{train}} - x_{\text{mean}}}{x_{var}} \quad (4)$$

From here, mean and standard variance vectors in (2) and (3) 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 PC, integrated an 8GB 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. Besides, these testing images are subjected to cover almost tampering methods in the whole formats of images (e.g., jpeg, png, tif, and bmp). 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:

$$\text{Reliability} = \frac{1}{k+1} \sum_{i=0}^{k} l(p_i) \quad (5)$$

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].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

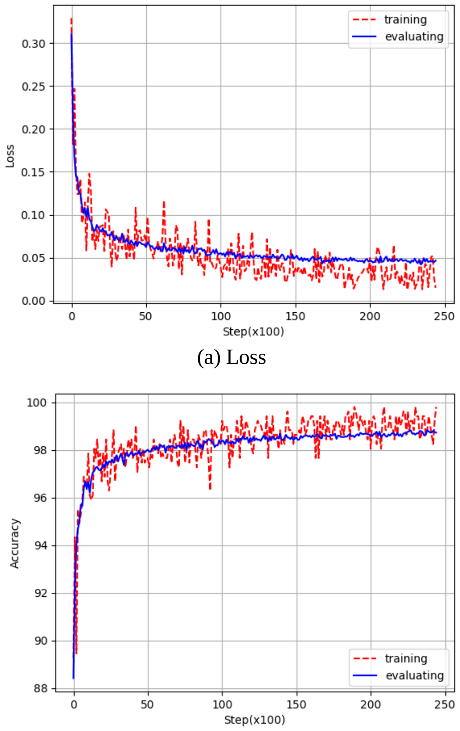
$$\text{Precision} = \frac{TP}{TP + FP} \quad (7)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (8)$$

$$F - \text{score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (9)$$

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, the *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, *F-score* is an answering one. In its formula, we can see that *F-score* contains information of both *Precision* and *Recall*. Besides, the range of *F-score* 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.

A. Training Process



(a) Loss

(b) Accuracy

Fig. 7. Loss and accuracy during the training process. These metrics are calculated on both of training set and evaluating set.

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. 7. 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 less than 5 minutes to train.

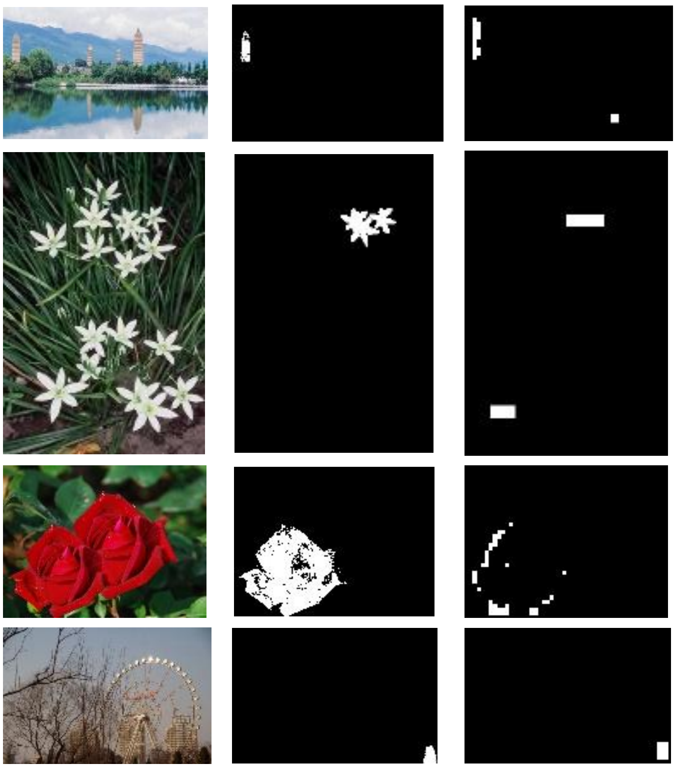
TABLE I: METRICS OF THE TRAINING AND TESTING PROCESS

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

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 32×32 window with stride of 16, overlapping patches are taken out from the image. Then, these RGB patches are converted into YCrCb, before being transformed in the Wavelet domain and fed into the trained 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. 8 draws testing results of some images. Here, there are totally four columns (e.g., origin, 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.



(a) Tampered (b) Ground truth (c) Prediction

Fig. 8. Testing predictions of some images. Tampered images, ground truths, and predictions are depicted in the first, second, and third columns, in turn.

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. 9), 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.

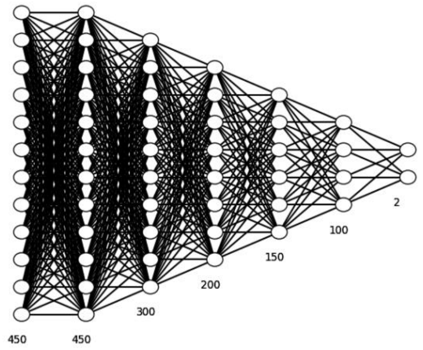


Fig. 9. A different neural network. This neural network is same as the original neural network in Fig. 2, excepting the number of neuron in the first layer. While the original one has 300 neurons, this neuron network has 450 neurons inputted to the first layer.

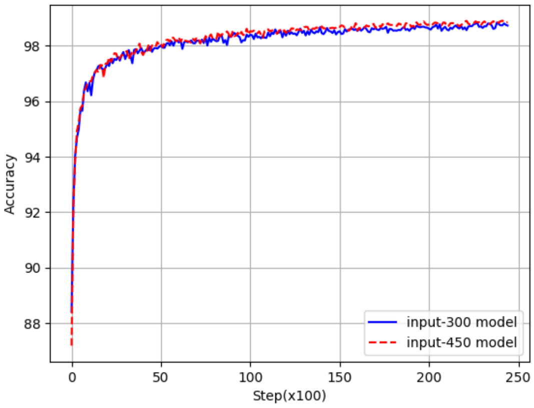


Fig. 10. Accuracies of two models during the training process. These metrics are calculated on the evaluating set.

Fig. 10 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 features for training are analytically selected in the way that positive and negative dataset are distinguished. Therefore, the input-300 model can learn robust features and avoid useless information from the extracted features. 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.

TABLE II: COMPARISON BETWEEN TWO NEURAL NETWORKS

Metric	Training		Testing	
	input-300	input-540	input-300	input-540
Accuracy	98.21%	98.68%	97.11%	96.92%
Precision	99.08%	99.00%	98.88%	99.38%
Recall	97.32%	98.31%	95.65%	94.87%
F-score	98.19%	98.65%	97.23%	97.07%
Time	240s	305s	3.57s	3.76s

D. Compare to Different Methods

After testing the proposed model, we continue comparing our performance to the others. To prove that data-driven can do better than analytical methods, two conventional ones [28], [29] are chosen, alongside with one more data-driven one [17]. This comparison uses detection accuracy metric obtained when testing on the CASIA-v2 database. Unit, which is used to calculate the detection accuracy, is image. 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 Convolutional Neural Network 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. Nonetheless, 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 (1052 neurons), comparing to the one of [17] (606752 neurons). 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 neurons
Rao *et al.* (2016) [17]	97.83%	606752
Proposed	**97.11%**	**1502**
Goh *et al.* (2015) [28]	96.21%	-
He *et al.* (2012) [29]	87.37%	-

V. CONCLUSION

In this paper, we have proposed a low computational-cost and effective data-driven model as a modified deep-learning based approach to solve the problem of Image Forgery Detection. A fully connected neural network, along with relating components (e.g., activations, initialization, normalization, optimizer), was designed to classify tampered patches. 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 learns 300-D features at the input 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 two conventional methods and a data-driven method of other authors. Our model can achieve a noticeable detection accuracy of 97.11%, while it must suffer tough conditions, namely narrowness in architecture and lack of positive data. In the conclusion, this model can show an effective approach for detecting forged images.

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 Trans. on Information Forensics and Security*, vol. 5, no. 4, pp. 857-867, 2010.

[2] I. Amerini, L. Ballan, R. Caldelli, A. D. Bimbo, and G. Serra, "A sift-based forensic method for copy-move attack detection and transformation recovery," *IEEE Trans. on Information Forensics and Security*, vol. 6, no. 3, pp. 1099-1110, 2011.

[3] P. Kakar and N. Sudha, "Exposing post-processed copy-paste forgeries through transform-invariant feature," *IEEE Trans. on Information Forensics and Security*, vol. 7, no. 3, 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," in *Proc. Information Hiding Conference, Lecture Notes in Computer Science*, Springer, Heidelberg-Berlin, 2010, vol. 6387.

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

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

[7] L.-T. Thuong, H.-K. Tu, P.-C.-H. Long, T.-H. An, D. Nilanjan, and L. Marie, "Combined zernike moment and multiscale analysis for tamper detection in digital images," *An International Journal of Computing and Informatics*, vol. 41, no. 1, March 2017.

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

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

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

[11] L. Thing, Y. Chen, and C. Cheh, "An improved double compression detection method for JPEG image forensics," in *Proc. IEEE International Symposium on Multimedia*, December 2012, pp. 290-297.

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

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

[14] C. Chang, C. Yu, and C. Chang, "A forgery detection algorithm for exemplar-based in-painting images using multi-region relation", *Journal Image and Vision Computing*, vol. 31, no. 1, 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, pp. 1849-1853, Nov. 2015.

[16] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *Proc. the 4th ACM Workshop on Information Hiding and Multimedia Security*, New York-USA, 2016, pp. 5-10.

[17] R. Yuan and N. Jiangqun, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *Proc. IEEE International Workshop on Information Forensics and Security (WIFS)*, Abu Dhabi-United Arab Emirates, 2016.

[18] J. Ouyang, Y. Liu, and M. Liao, "Copy-move forgery detection based on deep learning," in *Proc. 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics*, Shanghai, China, 2017.

[19] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Proceedings of the 25th Int. Conference on Neural Information Processing Systems*, Nevada-USA, 2012, vol. 1, pp. 1097-1105.

[20] Y. Zhang, J. Goh, L. Win, and V. Thing, "Image region forgery detection: a deep learning approach," in *Proc. the Singapore Cyber-Security Conf.*, Singapore, 2016.

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

[22] J. Dong and W. Wang, *Casia Tampering Detection Dataset*, 2011.

[23] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. the 13rd International Conference on Artificial Intelligence and Statistics*, Sardinia-Italy, 2010, pp. 249-256.

[24] V. Nair and E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc.* t*he 27th International Conference on Machine Learning*, Haifa-Israel, 2010, pp. 807-814.

[25] B. Xu, N. Wang, T. Chen, and M. Li. (2015). Empirical evaluation of rectified activations in convolutional network. [Online]. Available: https://arxiv.org/abs/1505. 00853v2

[26] K. He, X. Zhang, S. Ren, and J. Sun. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. [Online]. Available:https://arxiv.org/abs/1502.01852v1

[27] P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. 3rd International Conference for Learning Representations*, San Diego-USA, 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, pp. 76-104, March 2015.

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

Thuong Le-Tien was born in Saigon, Ho Chi Minh City, Vietnam. He received the bachelor and master degrees in electronics-engineering from Ho Chi Minh 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 EEE 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 170 scientific 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 Ho Chi Minh 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.

Thuy Nguyen-Chinh was born in Dong Nai, Vietnam. Currently, he is a senior student in the Honor Class of Electronics and Telecommunications, Electrical and Electronics Engineering Department, Ho Chi Minh City University of Technology, HCMUT, Vietnam. His research relates to apply machine learning and deep learning techniques to solve problems in computer vision, particularly in image forensics, biometrics, and autonomous robotics.

Thien Do-Tieu is a senior student in the Honor Class of Electronics and Telecommunications, Electrical and Electronics Engineering Department, Ho Chi Minh 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.