GENERAL PURPOSE IMAGE TAMPERING DETECTION USING CONVOLUTIONAL NEURAL NETWORK AND LOCAL OPTIMAL ORIENTED PATTERN (LOOP)

Ali Ahmad Aminu[1, 2] and Nwojo Nnanna Agwu[1]

[1]Department of Computer Science, Nile University of Nigeria, Abuja, Nigeria
[2]Department of Computer Science, Gombe State University, Gombe, Gombe, Nigeria

ABSTRACT

Digital image tampering detection has been an active area of research in recent times due to the ease with which digital image can be modified to convey false or misleading information. To address this problem, several studies have proposed forensics algorithms for digital image tampering detection. While these approaches have shown remarkable improvement, most of them only focused on detecting a specific type of image tampering. The limitation of these approaches is that new forensic method must be designed for each new manipulation approach that is developed. Consequently, there is a need to develop methods capable of detecting multiple tampering operations. In this paper, we proposed a novel general purpose image tampering scheme based on CNNs and Local Optimal Oriented Pattern (LOOP) which is capable of detecting five types of image tampering in both binary and multiclass scenarios. Unlike the existing deep learning techniques which used constrained pre-processing layers to suppress the effect of image content in order to capture image tampering traces, our method uses LOOP features, which can effectively subdue the effect image content, thus, allowing the proposed CNNs to capture the needed features to distinguish among different types of image tampering. Through a number of detailed experiments, our results demonstrate that the proposed general purpose image tampering method can achieve high detection accuracies in individual and multiclass image tampering detections respectively and a comparative analysis of our results with the existing state of the arts reveals that the proposed model is more robust than most of the exiting methods.

KEYWORDS

Image Tampering, General purpose Tampering Detection, Convolutional Neural Network, Local Optimal Oriented Pattern

1. INTRODUCTION

Today, digital media including images have become one of the main medium of communication due to their expressive abilities, ease of acquisition, distribution, and storage [1]. The effectiveness of digital images in conveying information have made them more preferable than text information as a means of communication. Consequently, it is becoming common more than ever to see an image representing a prime source of evidence in the court of law, a prime source of information in crime investigation, a source of news by mass media and news publishing agencies. At the same time, the nature of the digital image has raised a lot of questions in most of these positive aspects where they are employed. A digital image can be easily modified to convey false or misleading information. The development of sophisticated computers and image editing software has made the manipulation of digital images very easy [2]. Thus, the manipulations of images for malicious purposes are now rampant in our society leading to many ethical and moral

DOI: 10.5122/sipij.2021.12202

consequences, such as the spread of fake news, wrong verdict, and damage of reputation among others [3]. This difference between the importance of digital image on one hand and the uncertainties regarding their vulnerability to manipulations on the other hand calls for a reliable and efficient method of assessing their authenticity before basing important societal decisions on them. Hence, the study and research on detecting digital image tampering becomes vital in today's digital era.

In line with the above, a handful of image tampering detection and localization approaches have been proposed in recent times, aiming at improving the state of the art image tampering detection and localization methods. The earliest methods such as [4], [5], and [6] perform image tampering detection by exploiting frequency domain features, Color Filter Array features, and local binary descriptors. Although, these approaches have been successfully used to detect different type of image tampering, they could not be used to detect more than one type of image tampering operations because they operate based on the fact that each image tampering operation usually leave traces behind. To identify a particular image tampering operation, researcher designed algorithms that will extract these traces and use them to detect targeted image manipulation. The limitations of these approaches is that they are designed to detect only one image tampering type. Thus, several test must be carried out to detect whether a given image is tampered or authentic [3]. Consequently, there is a need to develop methods capable of detecting multiple tampering operations.

To address these problems, recent studies in image tampering have focused on designing general purposed or universal approaches capable of detecting more than one tampering type**.** Inspired by the performance of Spatial Rich Model (SRM) in image steganalysis, many general purpose image tampering detection approaches [7] [8] [9] [10] utilizing SRM features have been proposed, which yielded excellent results. With the success of deep learning methods, specifically, CNN in many visual recognition tasks, recent studies [11] [12] [13] [14] in image forensics also seek to leverage the strength of deep learning methods to solve the problem of detecting and localizing digital image forgery**.** These methods can automatically learn manipulations traces directly from data without the need for handcrafted features or human analysis by using a set of convolution kernels whose weights are learned via neural network training technique known as back-propagation. Hence, they provide better performance than the earliest methods. While these methods have improve on the performance of image tampering detection systems, the performances of these systems still requires significant improvement, so that they can meet the need of real-world forensic task.

Therefore, in this paper, we proposed a novel general purpose CNNs and LOOP based image tampering detection method capable of detecting different image tampering operations in both binary and multiclass classification scenarios. Unlike existing approaches that uses hand designed features or constrained pre-processing layers, the proposed method can directly extract image tampering traces directly from the LOOP images, which has the effect of suppressing the effect of image content allowing the proposed CNN to capture the different traces needed to detect different types of image tampering concurrently. To assess the performance of the proposed model, we have tested it through a number of experiments and the results of these experiments demonstrate the effectiveness of the proposed model in detecting individual, multiple, as well as manipulation chains image tampering in both un-compressed and compressed image formats.

The rest of this paper is organized as follows: Section 2 presents the related work. Section 3 discusses the proposed method. Section 4 presents the experiment and results. Finally, section 5 concludes the paper and highlights future research direction.

2. RELATED WORK

A handful of techniques have been proposed for detecting image tampering such as copy-move, image splicing, object removal, and content preserving manipulations such as median filtering and JPEG compression. In this section, we will briefly discuss some of the existing methods used for detecting multiple digital image tampering.

Inspired by the success of Spatial Rich Model (SRM) in many image steganalysis task, Qiu et al. [15], suggested the use of steganalytic features such as SPAM (Subtractive Pixel Adjacency Matrix) [16] and SRM [17] to detect six different types of image processing operations. The work of Fan et al. [18] used Gaussian mixture models to model the statistics of image process by different image operation. Bayar and Stamm, [3] presented a universal image manipulation detection technique that utilizes deep learning approach. They proposed a new convolution layer called constrained convolution layer capable of suppressing image content to learn image manipulation operation directly from data. Their method could effectively detect specific and multiple image manipulation operation and it showed superiority over Spatial Rich Model (SRM) based general purpose image manipulation approaches.

Moreover, Sundus et al [19] also investigate and demonstrate the performance of SRM and Local Binary Pattern (LBP) in detecting multiple image tampering. They embedded LBP in SRM sub-models to capture detailed statistics of the quantized version of image noise residuals. The resulting features were used for classification using an ensemble classifier. In [20], chen et al. proposed a new CNN based method to adaptively learn discriminative features for identifying image processing operations. They carefully designed the high pass filter bank to get the image residuals of the input image, the channel expansion layer to mix up the resulting residuals. Their method achieved state of the art result. The authors of [21], suggested a dual stream CNN model for detecting resampling of recompressed image. They used low-order high pass filter for computing image residuals used for classification. Zhan et al. [22] proposed a transfer learning based technique which enable them to train their model using small amount of training data. In [23], authors proposed a universal image forensics method based on Siamese network. The proposed method takes as input a pair of image patches and decides whether they are identically or differently processed.

Recently, the work of [24], Fridrich and Boroumond presented a model for detecting the processing history of image based on CNN with an IP layer accepting statistical moments of feature maps. The proposed model could correctly classify images of different size and is robust to JPEG compression. In [25], authors proposed a general purpose forgery detection and localization using anomalous features. Their approach works on image of arbitrary size and could detect and localize many known and unknown types of image forgery. In [26], chen et al. proposed a multipurpose image forensics tool using densely connected CNN that could detect 11 different types of manipulations. Their method works efficiently on image of different sizes and it was robust against JPEG compression. Zhang and Ni [27] proposed a dense Unit with a cross-layer intersection for detection and localization of image forgeries. They initialized the weights of their network with high pass filters used in SRM and used a multi-stage training approach to speed up convergence.

3. PROPOSED METHODOLOGY

In this section, we present an overview of the proposed approach for general purpose digital image tampering detection, followed by a detail description of the key aspects involved in the subsequent subsections. The proposed method utilizes the discriminative strengths of Local

Optimal Oriented Pattern (LOOP) [28] and the feature extraction capabilities of Convolutional Neural Networks (CNNs) to build a general framework for digital image tampering detection. Developing a general purpose image tampering detection approach requires the use of low-level features that can reveal fingerprints of different types of image tampering. LOOP is capable of suppressing the semantic information in an image allowing the proposed CNN to extract and learn the low level features required to distinguish different types of image tampering. Thus, unlike existing techniques that utilize actual image as input to their CNN model, we employed LOOP representation of images as the input of the proposed CNN design due to their discrimination strengths and the fact that image manipulation traces are more pronounced in LOOP image than the original image. Figure 3.1, gives an overview of the proposed method. First an RGB image is converted to grayscale image, from which the LOOP image is computed. The LOOP image having the same size as the original image is then fed to the designed CNN architecture which then extract and learn deep features that will be used for classification by the fully connected layers. To further assess the feature extraction capabilities of the proposed CNN, deep features extracted from the second dense layer of the proposed CNN are used to train and evaluate two tree based classifiers, XGBOOST and Extra tree classifier, on the same task performed by the proposed CNN.

3.1. Local Optimal Oriented Pattern (LOOP)

LOOP is a binary descriptor recently proposed by Chakraborti et *a*l [28], mostly used in the description of local image features and the classification of texture. Texture in this context refers to gray scale or color pixel intensities of image. LOOP is an enhancement of local binary descriptors such as Local Binary pattern (LBP) [29], and Local Directional Pattern (LDP) [30] and their variants which have recorded many success in image classification tasks including many image forensics tasks [19][31][32]. LOOP improved on existing local descriptors by adding rotation invariance into the main formulation of local descriptors, decreasing computational time and increasing overall accuracy [28].The texture of an image provides details about the grayscale or colour pixel intensities variation. Tampering operations such as, median filtering, copy move, image splicing etc. inevitably introduce textural variations in images. LOOP are powerful texture descriptors which can capture such variations caused by the tampering operations. To leverage on the feature extraction capabilities of CNNs and the discriminative strengths of LOOP, we proposed a novel image tampering approach that combines the strength of CNNs and LOOP features.

3.2. LOOP Computation

Given an image *I*, LOOP representation can be obtained using equation (1) and (2) below. Suppose i_c represent the intensity of the image I at pixel (x_c, y_c) and i_n (n = 0, 1… 7) is the intensity of a pixel in the 3×3 neighborhood of (x_c, y_c) ignoring the center pixel i_c . The 8 Kirsch masks are oriented in the direction of the 8 neighboring pixels i_n (n = 0, 1, 2… 7), giving a measure of the strength of intensity variation in those directions respectively.

Suppose $m_n$denotes the responses of the 8 Kirsch masks corresponding to pixels with intensity i_n (n = 0, 1, 2… 7). An exponential weight w_n(a digit between 0 and 7) is assigned to each of these pixels according to the rank of the magnitude of $m_n$among the 8 Kirsch masks activations.

Then, the LOOP value for the pixel(x_c, y_c)) can be computed as:

$$Loop(x_c, y_c) = \sum_{n=0}^{7} f(i_n - i_c).2^{w_n} \qquad (1)$$

Where

$$f(x)=\begin{cases}1, & if \quad x \geq 0 \\ 0, & otherwise\end{cases} \qquad (2)$$

The LOOP image is formed by accumulating the occurrence of the LOOP values over the entire image, i.e. each image is represented as a collection of LOOP values obtained from equation (1). Each of $i_n$neighbouring pixel oriented in the direction of the response of the 8 kirsch masks is compare with the central pixel $i_c$which evaluates to either 0 or 1 according to equation (2) forming an 8 digit binary number. Weights are then assigned to each of these binary numbers according to the rank of the magnitude of m_n among the output of the 8 Kirsch masks. The resulting binary number is then used as a label for that central pixel. For convenience, the binary number is converted to decimal. Repeating this procedure for the entire pixels of an image, the LOOP image is computed having the same size as the original image.

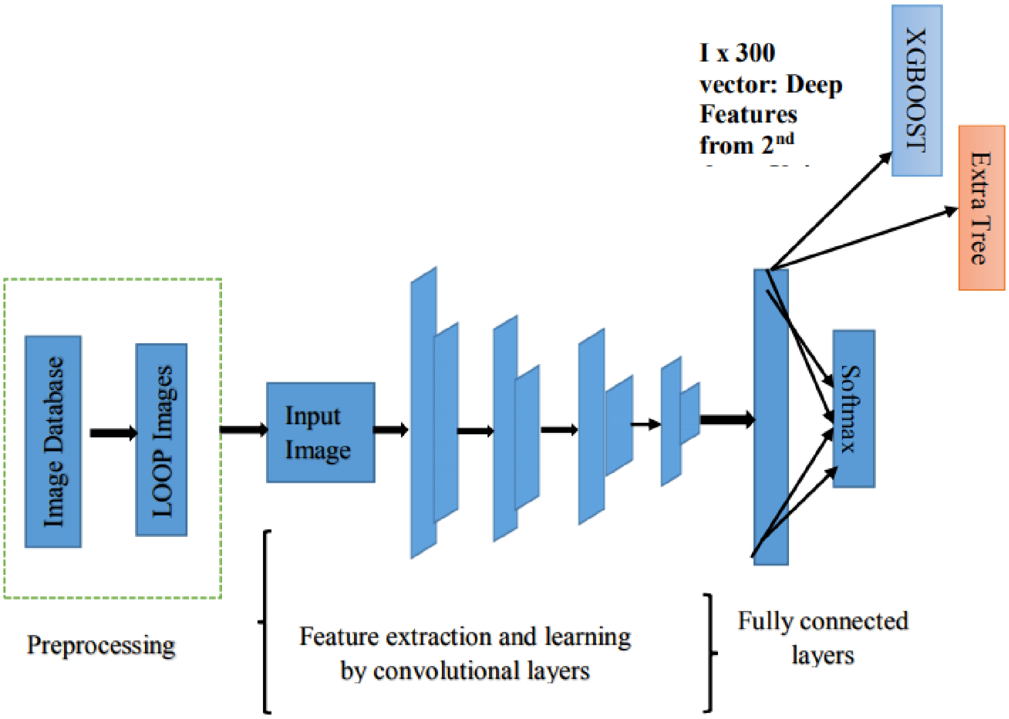


Figure 1: An overview of the proposed framework for general purpose image tampering detection. First, an input image is converted to LOOP representation which serves as input to the CNN network. Then the convolution layers of the network extracts and learns generic low level features from the LOOP images. The learned features are forwarded to the fully connected layers (dense units) which classifies an image as either tampered or not. To further validate the feature extraction capabilities of the proposed network, deep features from the second dense layer of the proposed network are extracted and used for training and testing of XGBOOST and extra tree classifiers on the same task performed by the proposed network.

The resulting LOOP images are used as input to our proposed model. Figure 3.2 illustrates the LOOP images of the original image and that of the four corresponding tampered image. Applying LOOP have successfully suppressed the effect of the image content, thus, allowing us to easily extract tampering traces introduced by different image tampering operation. The five images differs in their textural properties which shows that the different tampering operation affects the pixel intensity of an image differently. The LOOP images for the altered images have introduced noises of varying intensities and forms when compared to that of the original image. These noticeable differences in appearance among the five LOOP images signifies that the LOOP image has successfully capture the different texture variation induced by the different tampering operations.

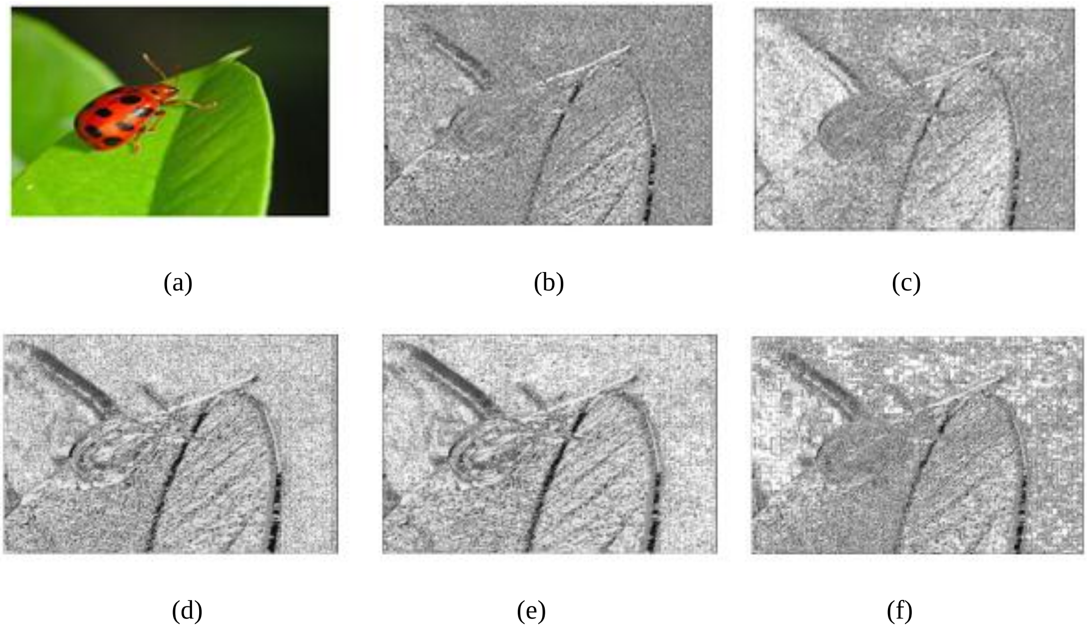


Figure 2: illustration of LOOP images of Original and Tampered images. (a) Original Image, LOOP image of (b) Original Image, (c) Gaussian Blurred Image, (d) gamma Corrected image (e) Median Filtered Image and (f) JPEG Compressed image.

3.3. The Proposed CNN Architecture for General Purpose Image Tampering Detection

Figure 3.3 illustrates the overall architecture of the proposed CNN for general purpose image tampering detection. The proposed CNN contains 13 layers, the first 10 are convolution layers and the last 3 are fully connected layers. Batch Normalization operation and activation function are disable in the first convolution layer to prevent the loss of information, since they are information losing operations. Similarly, layer 1 to 4 is directed connected without pooling operations. Pooling operations were only introduced after layers 5, 7, 9 and 10 respectively to reduce the size of the feature maps. The output of the second dense unit (FC 2) is then fed to the Softmax, XGBOOST and Extra Tree classifiers which respectively classify an image as either original or tampered. In the proposed CNN design, the input layer is a 256×256 and 128 grayscale LOOP images, and network parameters such as the number of filters, their size, and initial values are arrived at experimentally. Below, we describe some of the features of the new network architecture.

3.3.1. Weight Initialization

CNN with many layers are usually difficult to train due to their huge amount of parameters and the fact that their loss function is non-convex [73], however, their performance can be optimized by proper network initialization. In the proposed CNN design, we experimented with different weight initialization schemes namely Xavier initialization with normal bias and zero bias, He-normal with normal bias and zero bias, and lastly truncated normal with normal and zero bias. Experimenting with all methods of initialization leads to the realization that our model performs best when the weights are initialized using Xavier initialization and biases are initialized with zeros in all convolutional layers. Unlike other related studies such as [73], we did not constrain the parameters of our kernels to fixed high pass filters which may affect the generalization ability of models.

3.3.2. Convolution layers

The convolution operation uses one or several kernels to filter the input image, generating an array of feature maps for subsequent processing. The convolution operation is typically expressed by:

$$X_j^{l+1} = \sum X_i^l * W_{i\,j}^l + b_j^i \tag{3}$$

Where X_i^l denote the j^{-th}feature map of the i^{-th}layer, $W_{i\,j}^l$ is the trainable convolution kernels connecting the j^{-th}output map and the i^{-th}input map, b_j^i is the trainable bias parameter for the j^{-th} output map. As depicted in **Figure 3.3**, the proposed model uses 10 convolution layers. The first convolution layer "conv1" filters the 256×256×1 LOOP input image with 5 filters of size 7×7 using a stride of 1. The second convolution layer "conv2" filters the feature maps (250×250×5) produced by the first convolution layer with 144 kernels of size 5×5 using a stride of 2. 123×123×144 feature maps output by the second convolution layer is filtered by the third convolution layer "conv3" with 64 kernels of size 3×3 with a stride of 2 to yield a feature map of 61×61×64. BN and activation are disable in the first convolution layer and the first four convolution layers are connected directly without pooling layers to prevent information loss. These layers are responsible for extracting and learning image tampering fingerprints from the input images. The next seven convolution layers namely "Conv4" with 64 filters of size 3×3 and stride of 1, "Conv5" with 64 filters of size 3×3 and stride of 1, "Conv6" with 64 filters of size 3×3 and stride of 1, "Conv6" with 64 filters of size 3×3 and stride of 1, "Conv7" with 64 filters of size 3×3 and stride of 1, "Conv8" with 64 filters of size 3×3 and stride of 1, "Conv9" with 64 filters of size 3×3 and stride of 1, "Conv10" with 128 filters of size 3×3 and stride of 1, which are responsible for learning further discriminative features yielded the following output dimensions, 61×61×64, 30×30×64, 30×30×64, 15×15×64, 15×15×64, 7×7×64 and, 3×3×128 respectively.

3.3.3. Batch Normalization (BN)

The distribution of input data to internal layers changes as data flows through deep neural networks during training due to changes in the network parameters. This affects the learning capacity and accuracy of the network. To partially overcome this problem, we applied BN after each regular convolution layer of the proposed CNN design with the exception of the first convolution layer. The BN after each convolution layer normalizes elements in each feature map to zero (0) mean and unit (1) variance before feeding it to the next layer while training, hence accelerating training and increasing overall accuracy.

3.3.4. Activation Function

To introduce nonlinearity to neural networks, which greatly increases their capabilities of feature representation, a convolution layer is usually followed by a non-linear mapping known as an activation function. There is a variety of choice for an activation function such as tanh, sigmoid, and relu, etc. however, we tested the performance of the proposed model using various activation functions, combing more than one type of activation in some test cases, our investigation shows that the proposed model works best while using elu activation which can be represented mathematically by:

$$X_j^{l+1} = \boldsymbol{f}(X_i^l) \qquad (4)$$

Where $\boldsymbol{f}$(.) corresponds to the elu non-linear operation.

Similar to [73] and due to the fragile nature of feature maps extracted in early layers, our model only introduce activation functions after the second convolution layer. Layers 2-12 uses the elu activation and the softmax activation is applied after the last fully-connected layer (layer 13), which is responsible for mapping features learned by the last FC layer to a set of probability values, each corresponding to one class in the n+1 classes under consideration.

3.3.5. Pooling

Pooling operation is often applied after regular convolution layer in CNN to reduce the dimension of the feature maps and to obtain a more compact representation of the input data. Thus, reducing computational burden and chances of over-fitting. Max pooling and average pooling are the most widely used pooling operations in CNN design. Max pooling does not take into account all the activations within the pooling region, instead, it returns only the strongest activation within the region. Such kind of activation works best for sparse feature representation [33]. However, applying max-pooling in applications such as image forensics or steganalysis where the weak signals are the signal of interest may lead to the loss of the desired information. Therefore, throughout our proposed model, we used average pooling which takes into account all the activations within the pooling region. It is also worth pointing out that since pooling operations like non-linear activations are information losing processes [73], our CNN design delayed the introduction of pooling layers until after the fifth convolution layer as depicted in figure 3. Pooling operations suppresses the noise structures introduced by image manipulation operations, which are the signal of interest in image tampering detection, leading to poor performance. The Pooling operation used can be computed as:

$$X_j^{l+1} = \boldsymbol{pool}(X_i^l) \qquad (5)$$

Where X_i^l denote the j^{-th}feature map of the i^{-th}layer and ***pool*** (.) denote mean pooling operation

3.3.6. Implementation details of the Proposed General Purpose Image Tampering Detection Method

All experiments were carried out using Open CV, Matlab and Tensorflow 2 on a machine with NVidia GeForce GTX 1050 GPU with 8GB of memory. For training, we used Adam optimizer with a learning rate of 0.0001 and the default values for the moments (β1 = 0.9, β2 = 0.999, and ε = 10^{-7}). Xavier initialization was used to initialize the weights of layer 1 through 10 of the proposed model with their biases set to 0. The three fully connected layers are initialized with random numbers from a zero-mean Gaussian distribution with 0.01 standard deviation and their biases were set to 0.05. We trained the proposed model in each experiment for 100 epochs without shuffling of training data between epochs with a batch of 16 images for each training iteration to minimize the categorical cross-entropy loss function to obtain best set of parameters for the network.

Suppose θ is the parameter vector representing the weight vector corresponding to the image tampering detection task, the categorical cross-entropy loss can be computed as:

$$L(\theta) = \frac{-1}{M}\sum_{m=1}^{M}\sum_{n=1}^{N} 1(y^m = n)\log(y^m = n|x^m;\ \theta) \qquad (6)$$

Where M and N denote the total number of image samples and the number of classes, 1(.) represents an indicator function which equals 1 if m = n, otherwise 0. y^mandx^m correspond to the image label and the feature of the sample m. We minimizing the categorical cross-entropy loss using the Adam optimizer with all the training samples to learn the network's optimal set of parameters. Using these learned parameters, the network can predict whether a given image is tampered or not from the test samples.

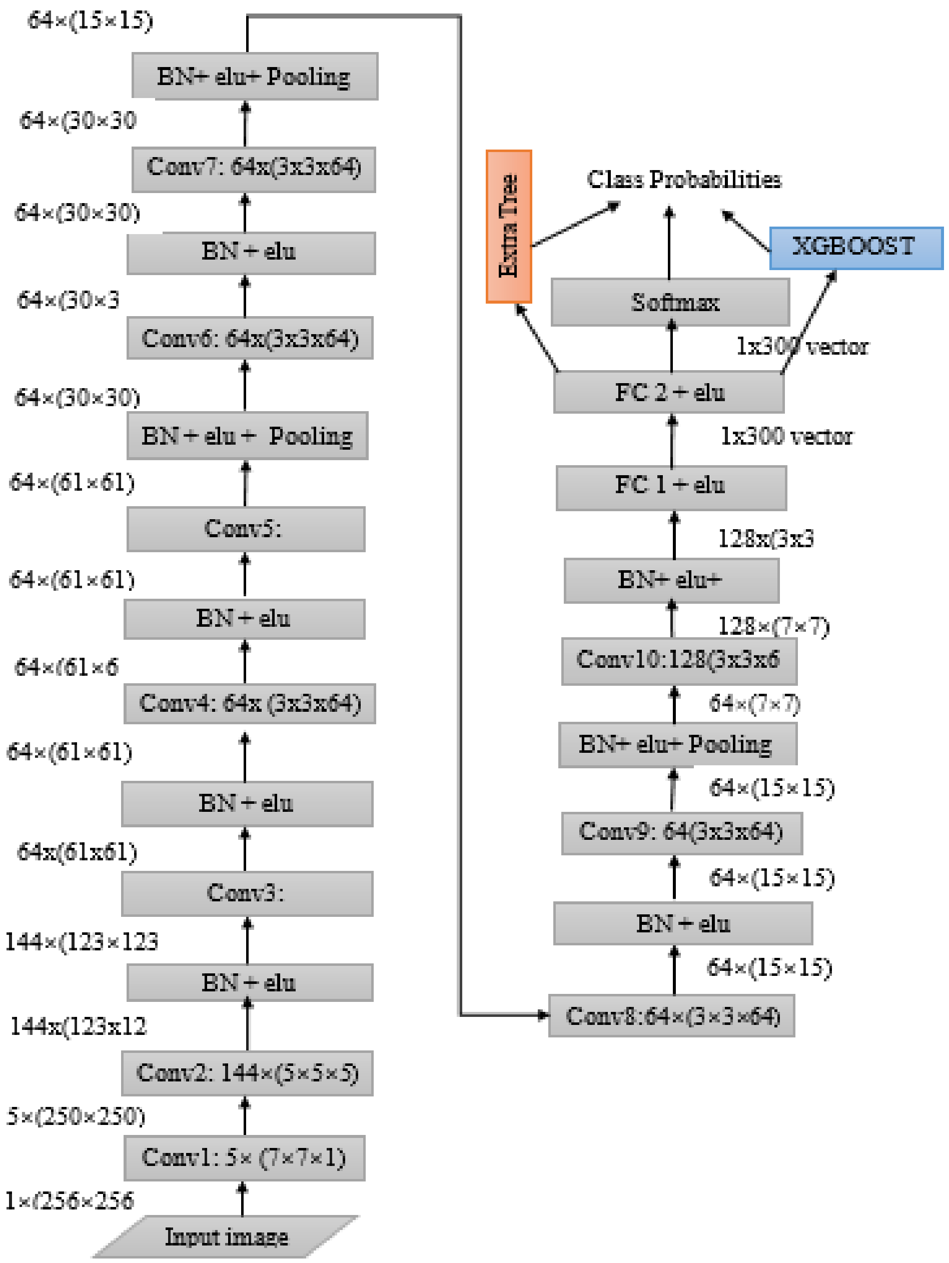


Figure 3: Architecture of the proposed CNN for General Purpose Image Tampering Detection

4. EXPERIMENTS AND RESULTS

A series of experiments was carried out to assess the performance of the proposed method in detecting image tampering operations. The first experiment evaluates the model's performance for detecting individual tampering operation, discriminating a real image from a tampered image. Here a binary classification was carried out for each of the tampering operations showed in table 3.1 against the original image. In the other experiments, the models ability to perform multi-class classification is evaluated with all the tampering operations listed in table 3.1. Next, we tested the model's performance on detecting multiple image tampering in images that have been subjected to JPEG compression. The impact of change in input image sizes and activation functions on the performance of the proposed model is also assessed. Finally, we compared the performance of the proposed model with some of the existing state of the arts image tampering detection methods from the literature. In the following subsections, we presents the proposed datasets as well as the results of the proposed model for the various image tampering scenarios considered in this section of the research.

4.1. Datasets

We consider five image tampering operations, Median Filtering (MF), Gaussian Blurring (GB), JPEG Compression (JC), Gamma Correction (GC) and Contrast Enhancement (EC) as shown in **Table 4.1**. We compare the performance of the proposed model with state of the art methods from the literature for both binary and multiclass classification on image datasets consisting of 9605 images obtained from the union of Microsoft Common Object in Context (MS COCO) [34], Bossbase datasets [35] and, IEEE image forensics [36] images databases. MS COCO is an image database widely used for object detection and image segmentation. IEEE images is an image database specifically created for image forensics challenges. The image database was formed by randomly choosing 3555 mages from MS COCO, 1050 image from IEEE and the remaining 5000 from Bossbase.

Table 1: Parameters used for creating experimental database

Tampering Operations	Parameters
Gaussian Blurring	K_{size}= 3×3, 5×5
Gamma Correction	ɣ = 0.5, 0.7, 0.8, 0.9
Median Filtering	K_{size}= 3×3, 5×5
Jpeg Compression	QF = 70, 80, 90
Contrast Enhancement	N/A

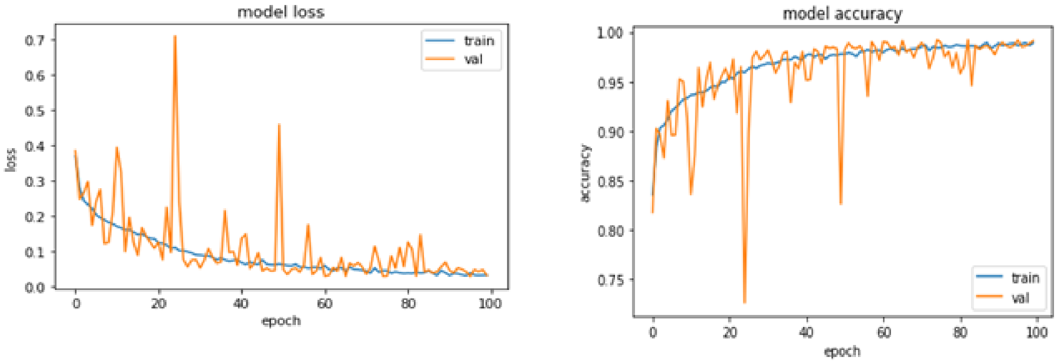
4.2. Experimental Results

4.2.1. Binaryclassification

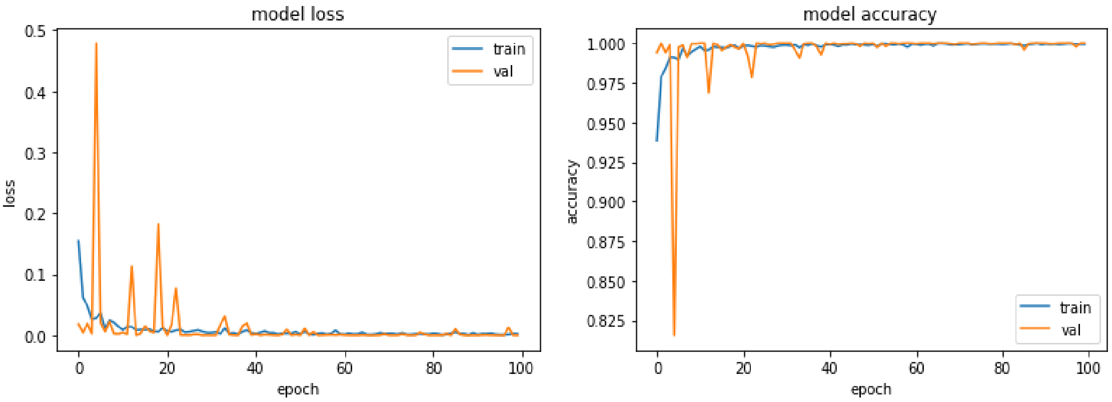
In the binary classification, the proposed model is used to detect whether a given image is original or tampered with one of the tampering operations listed in Table 1. Here the output layer of the proposed model contains two neurons, each corresponding to the original or tampered image respectively. Five experiments, each corresponding to that of one image tampering operation in Table 1 was conducted. To perform these experiments, for each of the 9605 original images, we generated 5 tampered versions using the tampering operations and the parameters listed in Table 1. Thus, we have five different database of images, each containing 9605 images and corresponding to one tampering operation in Table 1. We trained and evaluated the proposed binary classifier using 14000 and 2000 subset of the dataset respectively for 100 epochs using the implementation details of section 3.3.6 and kept the remaining 3000 for testing purpose. To further assess the richness and discriminative abilities of features extracted by the proposed CNN, we extracted deep features from the second dense layer of the proposed CNN and used it to train Extra Tree and XGBOOST classifiers for the same binary classification task. The three trained models were then used to predict the class labels for the testing images and the obtained accuracies are shown in Table 2. From these results, it can be observed that the proposed CNN can attain an accuracy of not less than 99.10 in detecting the different types of image tampering. The proposed CNN obtained the highest accuracies of 100% in the detection of Gaussian Blurring (GB) and JPEG compression (JPG). It can also be noticed that, with the Extra Tree based CNN, there was a slight increase or equal performance in the detection rates of each tampering when compared to that of the proposed CNN with the exception of GB detection in which the proposed CNN was better by 0.6%. Similarly, the XGBOOST based CNN classifier has improve the performance of the proposed CNN in GC and MF except in CE where the proposed CNN was better off by 0.07. Overall, The ET based CNN performed best in detecting individual image tampering operations among the three classifiers. Figure 4 shows the loss against epochs and accuracy against epochs plots of the proposed CNN for individual image tampering detection which further illustrate the performance of the proposed CNN model. These results indicate that the proposed CNN can effectively extract the needed image tampering traces for detecting different types of image tampering and converges just after few epochs.

Table 2: Proposed CNN, Extra tree and XGBOOST based CNNs detection accuracies (%) for individual image tampering detection

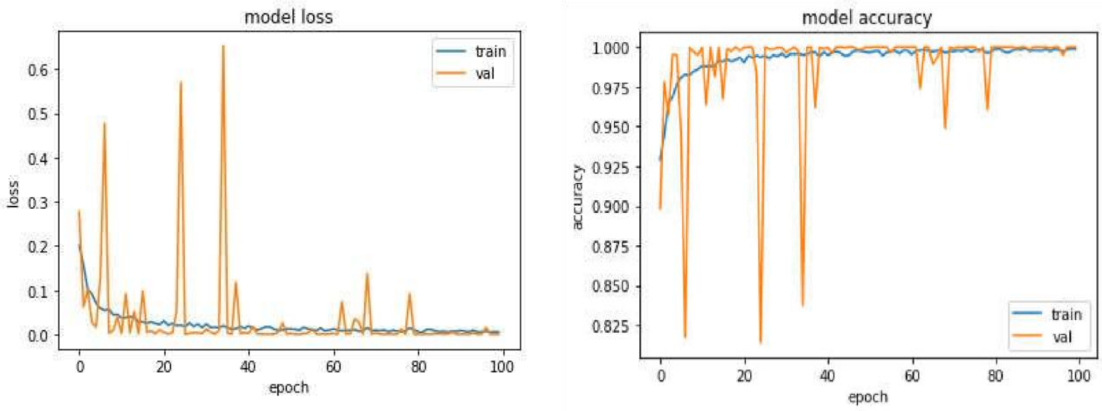
Classifiers	CE	GB	GC	MF	JPG
Softmax	99.10	100	99.90	99.46	100
Extra Tree Classifier	99.13	99.94	100	99.90	100
XGBOOST	99.03	100	99.92	99.94	100



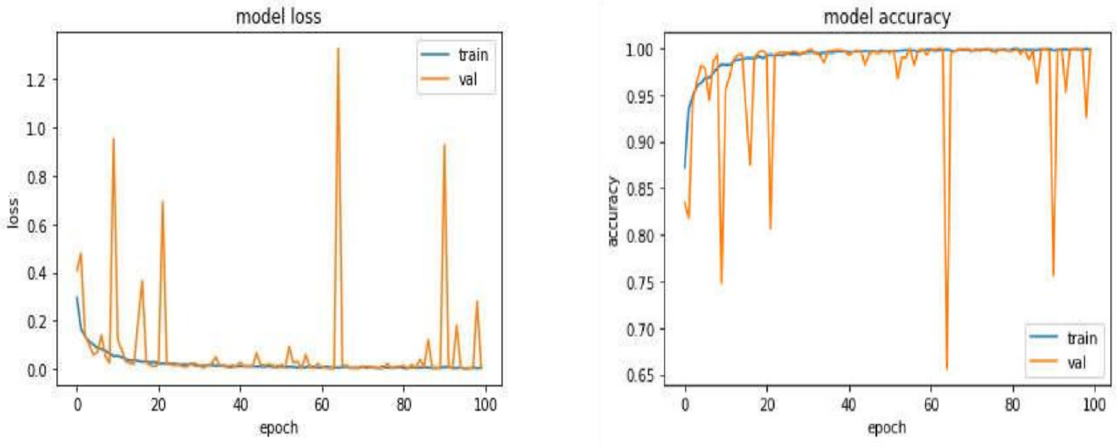
(a)


model loss
loss
0.5
0.4
0.3
0.2
0.1
0.0
0
20
40
60
80
100
epoch
train
val
model accuracy
accuracy
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
0
20
40
60
80
100
epoch
train
val

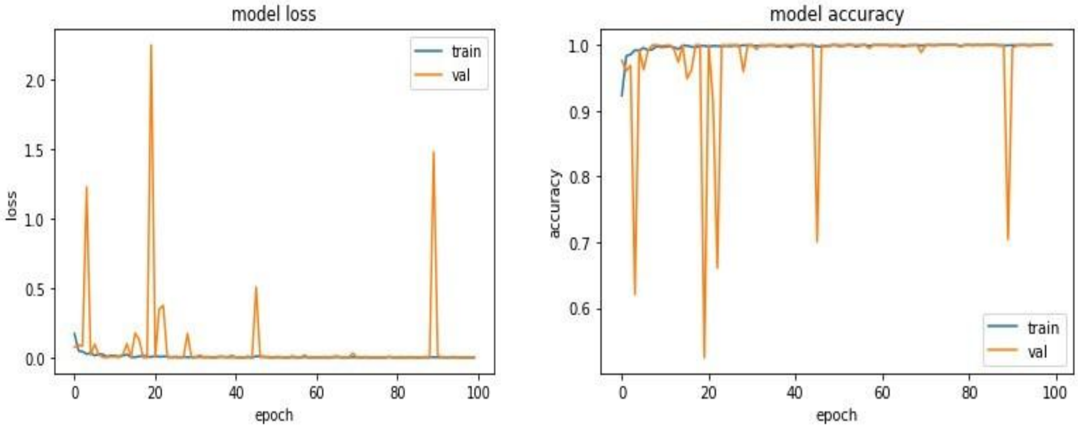
(b)


model loss
loss
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
20
40
60
80
100
epoch
train
val
model accuracy
accuracy
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
0
20
40
60
80
100
epoch
train
val

(c)


model loss
loss
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100
epoch
train
val
model accuracy
accuracy
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
20
40
60
80
100
epoch
train
val

(d)



(e)

Figure 4: Loss vs. epochs and accuracy vs. epochs of the proposed CNN for individual image tampering detection of (a) Contrast Enhancement (EC), (b) Gaussian Blurring (GB), (c) Gamma Correction (GC), (d) Median Filtering (MF) and (e) JPEG Compression (JC).

4.2.2. Comparison of the Proposed Binary Classifier against the State of the Arts

To further validate the performance of the proposed model on detecting individual image tampering, we compare the results of the proposed model on binary classification with the work of [2], [37], and, [38] from the literature on the same experiments. **Table 3** present the accuracies of the proposed model on individual image tampering in comparison with previous approach from the literature. The numbers highlighted in bold show the best result obtained for that tampering operation. The "-- "sign indicates the given method did not consider that tampering operations in evaluating their model.

Table 3: Comparison of the proposed model accuracies (%) for individual image tampering detection against the state of the arts

Methods	CE	GB	GC	MF	JPG
Proposed	99.13	**100**	**100**	99.94	**100**
Bayer [2]	--	99.95	--	99.71	99.66
Li [37]	**99.95**	**100**	96.76	**99.99**	99.94
Cozolina [38]	--	99.95	--	99.60	99.86

As shown in **Table 3**, it can be noticed that the proposed method improves the performance of [2] and [38] in GB, MF and JPG detections. Moreover, the result of the proposed model on GC showed that it has improve the performance of [90] by 3.24%. The only exception is in CE and MF where the method of [37] outperformed the proposed method by 0.08% and 0.05% respectively. These results demonstrate that the proposed method has better feature extraction and discrimination abilities, hence, it's more robust than the state of the arts in the majority of binary classification tasks performed.

4.2.3. Multiclass classification

In the previous subsection, the proposed model was used to construct a binary classifier which could effectively distinguish an original image from a tampered image. In this subsection, the proposed method is used to construct a multiclass classifier that could identify more than one image tampering type (all the tampering operations in Table 1). In this experiment, the output layer of the proposed model will have six neurons, one corresponding to the original image and the remaining five corresponding to the five tampering operations in Table 1 since it involves six classes. Thus, the main difference between the binary classifier and the multiclass classifier used in this work is the number of neurons each has in the output layer. To train the multiclass classifier, we randomly divided our image datasets into three disjoint subsets, with 7000, 1000 and 1500 images, respectively to create the training, validation and testing images. Next, for each image in the training set, we applied the five manipulations in Table 1 with their specified parameters and centrally crop each image to 256x256. Consequently, for each image in the training set, we now have 6 different types of images, the original image and its 5 tampered versions. After processing all the images in the training set, we ended up with 42000 (6 x7000) images in the training set. The LOOPs images of the training set were then computed and used as input for the proposed network. The same processing steps applied to the training set were replicated on the validation and test sets which generates 6000 validation and 9600 test sets used for evaluating and testing the model. The proposed model was then trained and evaluated for 100 epochs using the implementation details described in section 3.3.6.

Similar to the binary classification of section in the previous subsection, we extracted deep features from the second dense layer of the proposed CNN and used it to train and evaluated the performance of two Tree based models (Extra Tree and XGBOOST) for the same multiple image tampering detection. The three trained models were then used to predict the class labels for the testing image and the obtained results in terms of CM (Confusion Matrix) and average detection accuracies are reported in Figure 5 and Table 4 respectively. From these results, it can be noticed that the proposed CNN can achieve an average detection rate of 99.81% in multiple image tampering detection and can detect each tampering type with an accuracy of not less 98.27%. Table 4 also shows the average detection accuracy of extra tree based CNN and XGBOOST based CNN. Looking closely at the results, we can observe that each of the three classifiers could detect multiple image tampering with an average accuracy of at least 99.43%. However, unlike the binary classification, it can be noticed that the proposed CNN outperformed the extra tree based CNN and XGBOOST based CNN by 0.38% and 0.21% respectively. These results demonstrates the richness and discriminative strengths of the deep features extracted from the images by the proposed CNN model. Moreover, the results also indicate that the proposed CNN and LOOP model can effectively suppress the effect of image contents allowing it to extract the different image tampering traces needed for identification of various image tampering. Figure 6 presents the loss vs. epochs and accuracy vs. epochs of the proposed CNN for multiple image tampering detection, which further illustrate the performance of the proposed model.

Table 4: Average classification accuracies (%) of the proposed CNN, extra tree based CNN and, XGBOOST based CNN for Multiple image tampering detection

Proposed Method	Proposed CNN	Proposed CNN + extra tree Classifier	Proposed CNN + xgboost Classifier
Average Accuracies	99.81	99.43	99.60

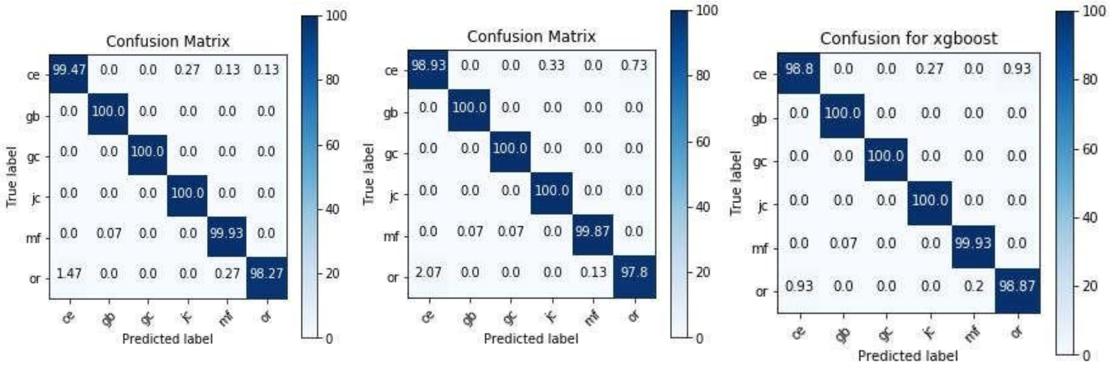


Figure 5: Confusion matrixes of the proposed CNN model, extra tree based CNN and, XGBOOST based CNN for Multiple image tampering detection

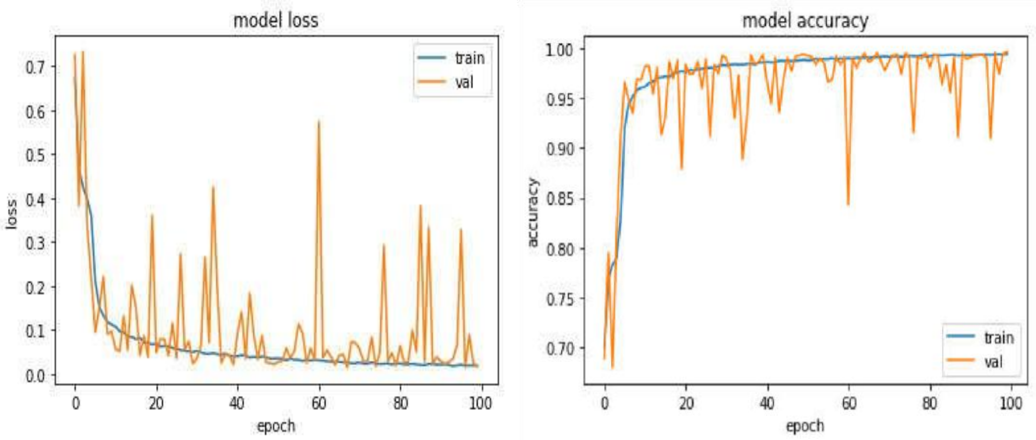


Figure 6: Loss vs. epochs and accuracy vs. epochs of the proposed CNN model for multiple image tampering detection

4.2.4. Robustness against JPEG Compression

As another example of the robustness of the proposed method, we assess its effectiveness against JPEG compression which is often used to conceal image tampering traces. To conduct this experiment, the processed training, validation and test sets of the images used in section 4.2.3 where JPEG compressed with two quality factors 70 and 90 respectively and the training, validation and testing of the models proceeds in a similar fashion to that of section 4.2.3 for the 70 and 90 JPEG compressed images respectively. Figures7 and 8 report the CMs of the three classifiers when they were applied for multiple image tampering detection in 70 and 90 JPEG compressed images respectively. The average detection accuracies of the three models for 70 and 90 JPEG compressed images in comparison to that of uncompressed images is provided in Table 5. It can be noticed from the results that the performance of the three classifiers decreased slightly in the JPEG compressed images. However, the proposed model could still detect all the image tampering with an accuracy of at least 97.13% for 70 and 97.20% for the 90 compression qualities respectively, indicating the robustness of the proposed model against JPEG compression. Although the proposed models performed excellently even in the JPEG compressed images, our results also indicate that JPEG compression has the effect of suppressing the different traces introduced by image tampering operations, thus, adversely affecting the overall performance of the proposed models.

Table 5: Average classification accuracies (%) of the proposed CNN, extra tree based CNN and, XGBOOST based CNN for multiple image tampering detection in uncompressed and compressed image formats.

Proposed Method	Proposed CNN	Proposed CNN + Extra Tree Classifier	Proposed CNN + XGBOOST
Uncompressed	99.81	99.43	99.60
70 compressed	99.01	99.08	99.21
90 compressed	98.93	98.72	99.24

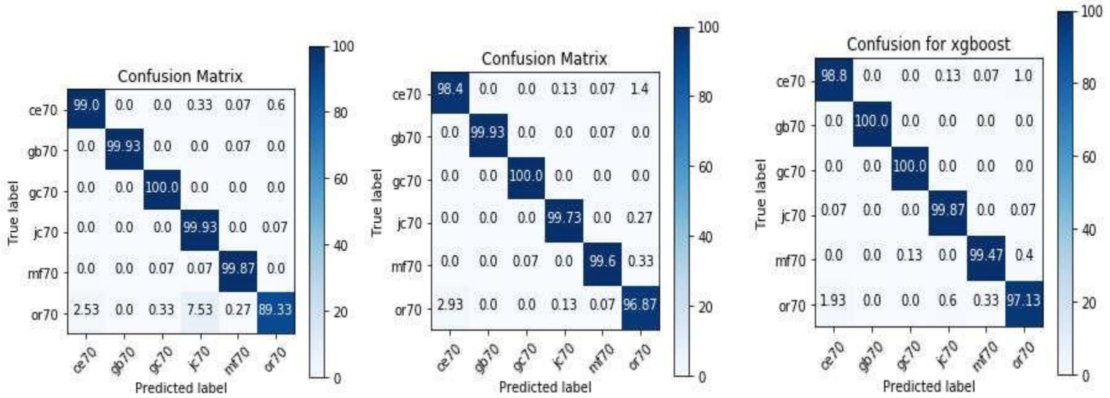


Figure 7: confusion matrixes of the proposed CNN model, extra tree based CNN and, XGBOOST based CNN for Multiple image tampering detection in compressed image (Q = 70)

4.2.5. Impact of Different input Image Size and Activation Functions

In the previous experiments, we have shown the robustness of the proposed model in detecting multiple image tampering in uncompressed and compressed images. In this subsection, we investigate the impact of different input image size and activation functions on the performance of the proposed models. To assess the impact of different input image size, we have trained the proposed models with two input image sizes i.e. 256 and 128 using the same experimental settings of **section 4.2.3** for multiple image tampering detection. The average detection accuracies obtained are reported in **Table 6**.

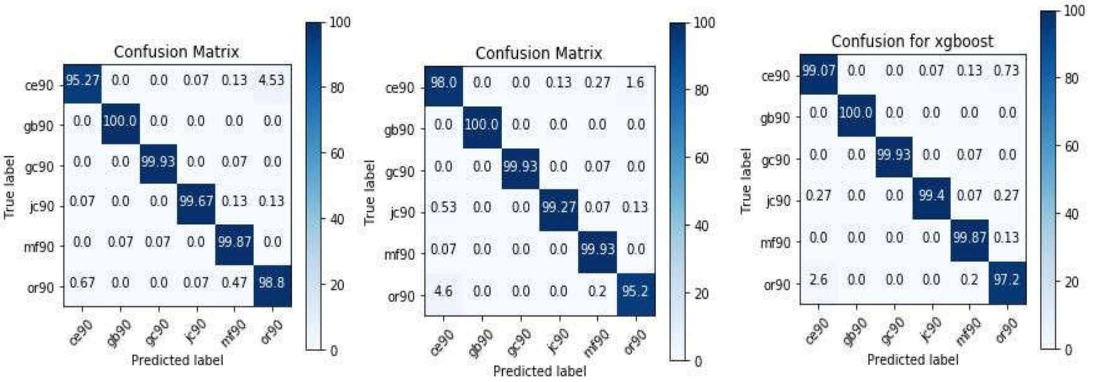


Figure 8: confusion matrixes of the proposed CNN model, extra tree based CNN and, XGBOOST based CNN for Multiple image tampering detection in compressed image (Q = 90)

From the results, we can notice that the average detection accuracies of the three models decreases with a decrease in the image size due to reduction in the statistical samples that could be extracted by the proposed CNN. However, the tree based CNN still performs excellently with an image size of 128. These results implies that, the choice of input image size may affect the performance of the proposed model. Larger input image size may improve performance at the expense of computational time, whereas smaller input image size may reduce computational time but at the expense of performance.

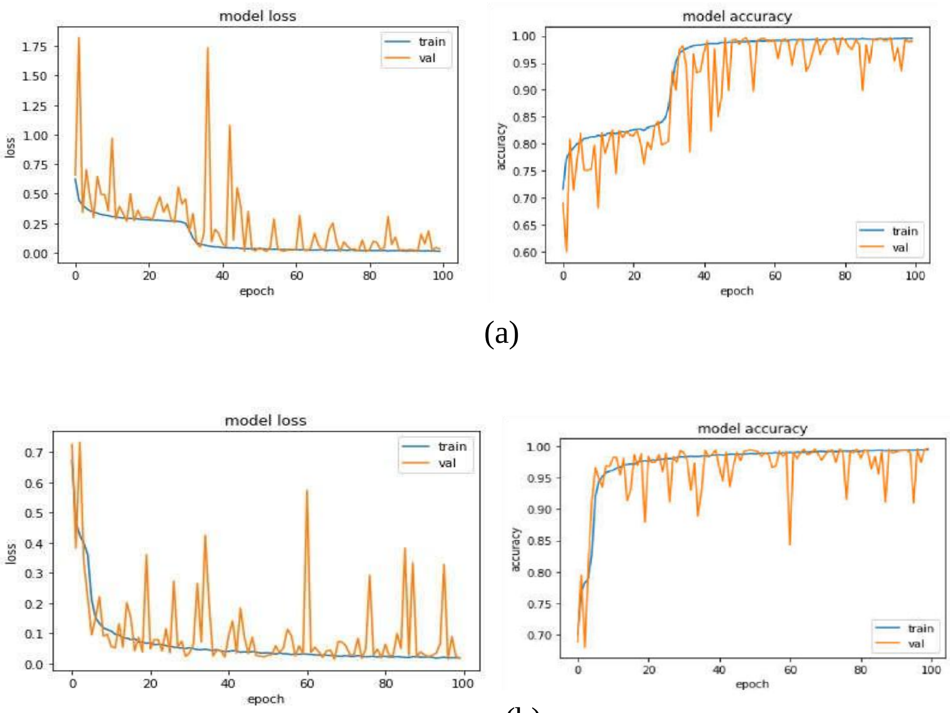
In another similar experiment, we investigate the impact of different activation functions on the performance of the proposed model for multiple image tampering detection. The proposed model was trained with two other activation functions (i.e. relu and tanh) using the same datasets and experimental settings of **section 4.2.3**. The results obtained by the relu and tanh based network in comparison with the proposed elu based network are provided in **Table 7**. From the results, we observed that the proposed elu network outperformed the relu and tanh based network by 0.25% and 0.47% respectively. **Figure 9** shows the loss and accuracy plots obtained by the tanh, elu and relu based networks respectively.

Table 6: Average classification accuracies (%) of the proposed CNN, extra tree based CNN and, XGBOOST based CNN for different input image sizes

Image size	256	128
Proposed CNN	**99.81**	97.54
CNN + Extra Tree	**99.43**	98.67
CNN +XGBOOST	**99.60**	99.28

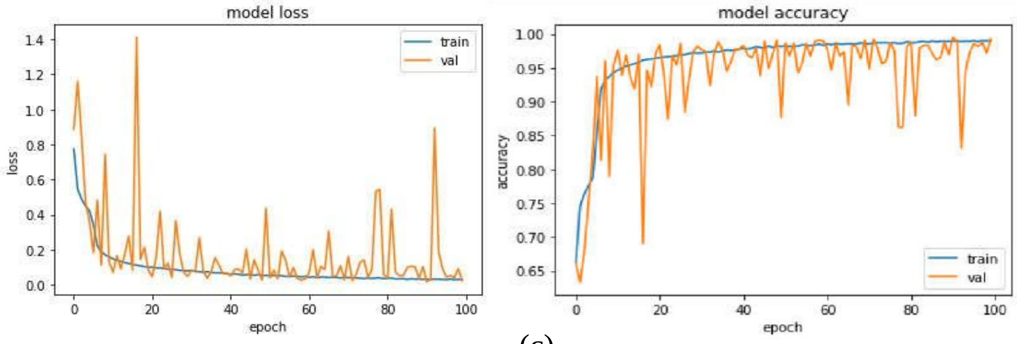
Table 7: Average classification accuracies (%) of the proposed CNN, extra tree based CNN and, XGBOOST based CNN for different activation functions

Activation	Tanh	Relu	elu
Proposed CNN	99.34	99.56	**99.81**
CNN + Extra Tree	99.43	99.31	**99.43**
CNN +XGBOOST	99.42	99.48	**99.60**



(a)

(b)



(c)

Figure 9: Loss vs. epochs and accuracy vs. epochs of the proposed CNN model for multiple image tampering detection using (a) relu, (b) elu and (c) tanh activation functions.

4.2.6. Limitation Analysis

Although the proposed image tampering detection method is robust against JPEG compression, our experiments revealed that JPEG compression may slightly degrades the performance the proposed model as it has the effect of suppressing the different traces introduced by image tampering operations. Similarly, from our experiments, it was also observed the performance of the proposed model decreases with a decrease in the image size due to reduction in the statistical samples that could be extracted by the proposed CNN. Larger input image size may improve performance at the expense of computational time, whereas smaller input image size may reduce computational time but at the expense of performance.

5. CONCLUSIONS

In this paper, we have proposed a novel general purpose CNN and LOOP based image tampering detection method. Unlike existing approaches that use hand designed features or constrained pre-processing layer, the proposed method can directly extract image tampering traces directly from the LOOP images, which has the effect of suppressing the effect of image content allowing the proposed CNN to capture the different traces needed to detect different type of image tampering. To assess the performance of the proposed model, we have tested it through a number of experiments and the results of these experiments demonstrate the effectiveness of the proposed model in detecting individual as well as multiple image tampering. To further assess the performance of the proposed model, we have compared its results with that of the state of the arts from the literature and these results also show that the proposed model could achieved a competitive performance. In future, we plan to extend the proposed method to detect anti-forensics image tampering operations as well as more image tampering operations.

ACKNOWLEDGEMENTS

This research was supported by National Information Technology Development Agency (NITDA) of Nigeria.

REFERENCES

[1] Redi, J. A., Taktak, W., and Dugelay, J. L., (2011) “Digital image forensics: a booklet for beginners”,Multimedia Tools and Applications, 133-62.

[2] Bayar, B., and Stamm, M.C., (2018) "Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection", IEEE Transactions on Information Forensics and Security 13, no. 11, 2691-2706.

[3] Bayar, B., and Stamm, M.C., (2016) "A deep learning approach to universal image manipulation detection using a new convolutional layer", In Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, 5 -10.

[4] Baby, L., and Jose, A., (2014) "Detection of Splicing in Digital Images Based on Illuminant Features", International Journal of Innovation and Scientific Research, 11(2351-8014), p.4

[5] Li L. L., Xue, J. Wang, X., and Tian, L., (2013) "A robust approach to detect digital forgeries by exploring correlation patterns", Pattern Anal Appl 1–15.

[6] Carvalho, T., Faria, F. A., Pedrini, H., Da R., Torres, S., and Rocha, A., (2016) ''Illuminant-based transformed spaces for image forensics'', IEEE Trans. Inf. Forensics Security, vol. 11, no. 4, pp. 720–733.

[7] Qiu, X., Li, H., Luo, W., and Huang J., (2014) "A universal image forensics strategy based on steganalytic model", In Proceedings of the 2nd ACM workshop on Information hiding and multimedia security, (pp. 165-170), ACM (2014 Jun 11).

[8] Sundus, F., Yousaf, M. H., and Hussain, F., (2017) "A generic passive image forgery detection scheme using local binary pattern with rich models", Computers & Electrical Engineering 62 (2017): 459-472.

[9] Rao, Y., and Ni, J., (2016) "A deep learning approach to detection of splicing and copy-move forgeries in images", In IEEE International Workshop on Information Forensics and Security (WIFS) (pp. 1-6), IEEE.

[10] Yue, W., AbdAlmageed, W., and P. Natarajan, P., (2019) "ManTra-Net: Manipulation tracing network for detection and localization of image forgeries with anomalous features", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9543-9552.

[11] Zhou, P., Han, X., Morariu, V. I., and Davis, L. S., (2018) "Learning rich features for image manipulation detection," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1053-1061.

[12] Boroumand, M., and Fridrich, J., (2018) "Deep learning for detecting processing history of images," Society for Imaging Science and Technology.

[13] Chen, Y., Xiangui, K., Yun, Q. S., and Jane Wang, Z., (2019) "A multi-purpose image forensic method using densely connected convolutional neural networks," Journal of Real-Time Image Processing 16, no. 3 pp. 725-740.

[14] Zhan, Y., Chen, Y., Zhang, Q., and Kang, X., (2017) "Image forensics based on transfer learning and convolutional neural network," In Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security (pp. 165-170).

[15] Qiu, X., Li, H., Luo, W., and Huang, J., (2014) "A universal image forensic strategy based on steganalytic model," in Workshop on Information hiding and multimedia security, ACM, pp. 165–170.

[16] Pevny, T., Bas, P., andFridrich, F., (2010) "Steganalysis by subtractive pixel adjacency matrix," IEEE Transactions on Information Forensics and Security, vol. 5, no. 2, pp. 215–224.

[17] Fridrich J., and J. Kodovsky, J., (2012) "Rich models for steganalysis of digital ` images," IEEE Transactions on Information Forensics and Security, vol. 7, no. 3, pp. 868–882.

[18] Fan, W., Wang, K., and Cayre, F., (2015) "General-purpose image forensics using patch likelihood under image statistical models," in IEEE International Workshop on Information Forensics and Security, pp. 1–6.

[19] Sundus, F., Yousaf, M. H., and Hussain. F. (2017) "A generic passive image forgery detection scheme using local binary pattern with rich models", Computers & Electrical Engineering 62, 459-472.

[20] Chen, B., Li, H., and Luo, W. (2017) "Image processing operations identification via convolutional neural network", arXiv preprint arXiv: 1709.02908.

[21] Cao, G. Zhou, G. A., Huang, X. Song, G. et al, (2019) "Resampling detection of recompressed images via dual-stream convolutional neural network," arXiv preprint arXiv: 1901.04637.

[22] Zhan, Y., Chen, Y., Zhang, Q., and Kang, X., (2017) "Image forensics based on transfer learning and convolutional neural network", In Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security (pp. 165-170).

[23] Mazumdar, A., Singh, J., Tomar, Y.S., and Bora, P. K., (2018) “Universal Image Manipulation Detection using Deep Siamese Convolutional Neural Network”,arXiv preprint arXiv: 1808.06323.

[24] Boroumand, M. and Fridrich, J., (2018) “Deep learning for detecting processing history of images”, Society for Imaging Science and Technology.

[25] Wu, Y., Abdalmageed, W., and Natarajan, P., (2019) “ManTra-Net: Manipulation Tracing Network for Detection and Localization of Image Forgeries With Anomalous Features”, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 9543-9552).

[26] Chen, Y., Xiangui, K., Yun, Q. S., and Jane Wang, Z., (2019) "A multi-purpose image forensic method using densely connected convolutional neural networks", Journal of Real-Time Image Processing 16, no. 3 pp. 725-740.

[27] Zhang, R., and Ni, J., (2020) "A Dense U-Net with Cross-Layer Intersection for Detection and Localization of Image Forgery", In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2982-2986.

[28] Chakraborti, T., McCane, B., Mills, S., and Pal, U., (2018) “Loop descriptor: Local optimal-oriented pattern”,IEEE Signal Processing Letters, 25(5), pp.635-639.

[29] Ojala, T., Pietikinen, M. and Harwood, D., (1994) “Performance evaluation of texture measures with classification based on Kullbackdiscrimination of distributions”, In Proc. ICPR.

[30] Jabid, T., Kabir, M. H. and Chae, O.S., (2010) “Gender Classification using Local Directional Pattern (LDP)”,In Proc. ICPR.

[31] RemyaRevi K., and Wilscy, W., (2020) “Image forgery detection using deep textural features from local binary pattern map”, Journal of Intelligent & Fuzzy Systems, (Preprint), pp.1-11, 2020.

[32] Surbhi, S. and Ghanekar, U., (2019) "Spliced Image Classification and Tampered Region Localization Using Local Directional Pattern", International Journal of Image, Graphics & Signal Processing 11, no. 3.

[33] Qian, Y., Dong, J., Wang, W. and Tan, T., (2015) “Deep learning for steganalysis via convolutional neural networks”, In Media Watermarking, Security, and Forensics 2015International Society for Optics and Photonics, Vol. 9409, p. 94090J.

[34] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al... (2014) “Microsoft COCO: Common objects in context”, InECCV, 2- 5.

[35] Bas, P., Filler, T. and Pevny T., (2011) “Break our steganographicsystem: The ins and outs of organizing boss,” In Information Hiding, pp. 59-70.

[36] IEEE IFS-TC image forensics challenge. IEEE Trans Inf Forensics Secur 2013.

[37] Li, H., Luo, W., Qiu, X. and Huang J., (2016) “Identification of various image operations using residual-based features”, IEEE Transactions on Circuits and Systems for Video Technology, 28(1), pp.31-45.

[38] Cozzolino, D., Giovanni, P. and Luisa V., (2017) "Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection", In Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security, pp. 159-164.