information

MDPI

Article

Optimization of a Pre-Trained AlexNet Model for Detecting and Localizing Image Forgeries

Soad Samir *, Eid Emary, Khaled El-Sayed and Hoda Onsi

Department of Information Technology, Faculty of Computers and Artificial Intelligent, Cairo University, Giza 12613, Egypt; eid.emary@aou.edu.eg (E.E.); k.mostafa@fci-cu.edu.eg (K.E.-S.); h.onsi@fci-cu.edu.eg (H.O.)
* Correspondence: s.samir@fci-cu.edu.eg or soad.samir@hotmail.com

Received: 20 April 2020; Accepted: 13 May 2020; Published: 20 May 2020

Abstract: With the advance of many image manipulation tools, carrying out image forgery and concealing the forgery is becoming easier. In this paper, the convolution neural network (CNN) innovation for image forgery detection and localization is discussed. A novel image forgery detection model using AlexNet framework is introduced. We proposed a modified model to optimize the AlexNet model by using batch normalization instead of local Response normalization, a maxout activation function instead of a rectified linear unit, and a softmax activation function in the last layer to act as a classifier. As a consequence, the AlexNet proposed model can carry out feature extraction and as well as detection of forgeries without the need for further manipulations. Throughout a number of experiments, we examine and differentiate the impacts of several important AlexNet design choices. The proposed networks model is applied on CASIA v2.0, CASIA v1.0, DVMM, and NIST Nimble Challenge 2017 datasets. We also apply k-fold cross-validation on datasets to divide them into training and test data samples. The experimental results achieved prove that the proposed model can accomplish a great performance for detecting different sorts of forgeries. Quantitative performance analysis of the proposed model can detect image forgeries with 98.176% accuracy.

Keywords: convolutional neural networks; AlexNet; activation function; forgery detection; batch normalization

1. Introduction

Motivated by the massive use of social media e.g., Facebook, Instagram, and Twitter, etc. and enhancements in image processing software applications, image forgery has become very popular and hence the need for image forgery detection has also increased.

Image manipulations that are done by the procedure of clipping and pasting areas, are one of the most well-known forms of digital image editing. This manipulation is distinguished as a copy–move image forgery. Image splicing is the most well-known type of image faking. It cuts and pastes areas from one or more different images cautiously to produce new synthesized digital images as shown in Figure 1. Therefore, detection and localization of these forgeries to reliably and automatically determine the authenticity of images have become an important and popular issue.

Figure 1. A fake image is created by splicing together content from two different images [1].

Recently, deep learning interest has grown and various noteworthy results are becoming visible. By this motivate, tampering detection researchers have attempted the use of deep learning to detect the images' changes without human intervention. Deep learning has been convenient in the field of image processing science. Two crucial areas are driving the success of deep learning use in image processing:

1. First, convolution neural network (CNN) architecture takes the fact that pixels and their neighborhood are highly correlated. Therefore, a CNN does not use one-to-one links among all pixels (as in major neural networks).
2. Second, CNN architecture counts on feature sharing, and so each channel or feature map is formed from a convolution operation using the same kernel at all positions [2].

The manipulation and editing of digital images has become a significant issue nowadays. There are various applications such as digital forensics, scientific publications, medical imaging, journalism, insurance claims, political campaigns, where image manipulation can be easily made. To specify whether an image is genuine or forged is a major challenge to researchers. The detection models proposed are beneficial to many applications in which the authenticity of a digital image has an influential impact.

Additionally to this, there are numerous editing processes executed on the forged areas to appear similar to the genuine areas. This demands the development of a universal forgery detection model that not only detects various image editing manipulations present in the forged image, but also can be capable of being generalized to editing manipulations not present in the forged image. This will let the model be more generalized to detect any type of editing or manipulations even if the model is not trained on it. The majority of the existing forgery detection models focalize on identifying a particular forgery editing (e.g., copy–move or splicing). Therefore, these models cannot perform better for other kinds of forgery. Additionally, it is impracticable and unrealistic to suppose that manipulation editing will be known in advance. In real-life, an image forgery detection model should be able to detect all types of manipulation editing rather than focalizing on a certain type.

Therefore, some questions exist with the account to CNNs design and training for image forgery detection:

- Do design parameters like the pooling mechanism or activation function choice have considerable effects on the accuracy?
- What effect do various normalization techniques like batch normalization and local contrast normalization have on CNN's accuracy?

To lead the research for using CNN models in image security, it is remarkable to address these issues. In this paper, we consistently analyze CNN design choices for image forgeries detection. Specifically, we investigate:

1. The effect of activation functions selection on the performance.

2. The effect of different normalization approaches such as batch normalization and local contrast normalization.
3. The variation between softmax classifier and SVM classifier.

Besides that, we prove that CNN can be designed to carry out several diverse forensic issues. The investigation done reveals that both general CNN design principles that are important regardless of the forensic assignment, along with other design choices that must be appropriately selected depending on the chosen forensic assignment. To ensure that the proposed model is robust, k-fold cross-validation is implemented, which means that the training process and testing process are executed on varieties of datasets that have been collected separately. The major contributions of the work done in this paper are as indicated in the following:

1. We propose an AlexNet model that is capable of detecting various image tampering and manipulations.
2. We introduce the proposed modified AlexNet model architecture, provide a detailed discussion of how it is constructed, as well as provide intuition into why it works.
3. We conduct a large scale experimental evaluation of the proposed architecture and show that it can outperform existing image manipulation detection techniques, can differentiate between multiple editing operations even when their parameters change, can localize fake detection results, and can provide excessively accurate forgery detection results when trained using a huge training dataset.

The motivation and reason behind choosing AlexNet as a core of the proposed model are that the ability of fast network training and its capability of reducing overfitting. The reasons why the AlexNet model is suitable for the analysis of forged images are its deep structure, its simple structure, fast training time, and less memory occupation. Provided that, the improvements we have made to the model (using max-out and batch normalization). All of these reasons lead the AlexNet to be one of the best choices in the forgery detection process. Through experiments sequence, the proposed AlexNet model can be learned automatically to discover and detect multiple types of image editing. This eliminates the need for time-consuming human intervention to outline forensic detection features. AlexNet is used to make the training faster and reducing overfitting. The remainder of this paper is organized as follows: Section 2 discusses the related works and gives an overview of how to use CNN in image forgery detection. Section 3 presents our study to obtain robust image manipulation and our framework to detect image forgeries. Section 4 shows our experimental results; and we conclude this paper in Section 5.

2. Related Work

In the latest years, techniques based on deep learning have become assertive. Some early work proposed CNN architectures with the first layer of high-pass filters, either fixed [3], [4] or trainable [5], meant to extract feature maps. It has been shown in [6] that successful methods based on handcrafted features can be recast as CNN and fine-tuned for improved performance. In Ref. [7] these low-level features are augmented with high-level ones in two-stream CNN architecture. In both [8,9], it was clarified that the constrained first layer used is better only for small networks and datasets. Given a reasonable large training dataset, deep models provide the identical results in favorable cases, but ensure higher robustness to compression and misalignments of training/test.

Several papers, beginning with paper [4] and followed by more recent papers [10] and [11], train the network to distinguish between homogeneous and heterogeneous patches which are known by the presence of both genuine and forged spaces. The case is to catch the features that describe transition regions, which are abnormal with respect to the background, to localize forgeries. This idea is followed also in [12], where the hybrid CNN-LSTM (long short term memory) model is trained to generate a binary mask for forgery localization. These methods, although, require ground truth maps to train the network, which may not be available.

For architectonic constraints, most of the methods perform a patch-based analysis, functioning on reasonably small patches, with additional steps needed to calculate a global outcome at the image-level. In Ref. [3], for example, CNN extracted features patch-wise and later aggregates them in a global feature vector used to feed an SVM (support vector machines) classifier. A major limitation is the need for large training and test datasets. Some methods, for example [5,11], use only one database and are split into groups of training and test; others [5] require fine-tuning on the target data. Such models and its procedures prove that the supervised learning generalization ability is shortened and limited.

Bayer and Stamm studied image manipulation detection by adding a new convolution layer [5]. Accordingly, CNN used a convolutional layer to identify the structural relationships among pixels anyhow of the image content. This model learned automatically how to detect image editing without relying on preprocessing or specific features. The model gave a high detection rate when only one of these specific attacks were implemented: median filtering, Gaussian blurring, additive white Gaussian noise, or resampling. If any other manipulations editing was applied to the forged image, this model failed and gave a bad detection rate.

Choi et al. studied CNN-based multi-operation detection to detect multiple attacks, not just only one attack [12]. Their technique proposed three types of processing, that have occurred repeatedly during image manipulation and were identified when they are applied to images. The model was convenient enough to detect these three manipulations. It can only solve three types of editing (GB: Gaussian blurring, MF: median filtering, GC: gamma correction). If this model applied on any different manipulations, it would give a low detection rate.

Salloum et al. [10] used a fully convolutional network (FCN) instead of CNN to locate the spliced regions. It classified each pixel in a spliced image as spliced or authentic. Two output branches of multi-task FCN are used to learn the labels and the spliced regions' edges respectively, and the two branches intersection output is considered to be the localization result. The model was evaluated on images from the Carvalho, CASIA v1.0, Columbia, and the NIST Nimble Challenge 2016 datasets. This model can solve splicing problem only with maximum F1 score 0.6117 on the Columbia dataset, and maximum MCC score 0.5703 on NIST 2016 dataset, which are very low to be used in the real-life problems.

In Ref. [3], the model applied max-pooling technique to the feature maps. The model consisted of 8 convolutional layers, three pooling layers and one fully-connected layer with a softmax classifier. They applied the framework on the public CASIA v1.0, CASIA v2.0 and DVMM datasets. The model used the SRM (spatial rich model) as a weight initialization instead of a random generation. SRM helps to improve the generalization ability and accelerate the convergence of the network. Major SRM problems can be listed as: it arises overfitting in some cases, increasing the processing time, and may other problems that lead the framework to unwanted results. This framework has another disadvantage is the rectified linear unit (ReLU) implementation as an activation function in the network. ReLU units can be fragile during training and can "die" which of course gives disappointing results.

Jaiswal, A. et al. [13] proposed a framework based on a combination of pre-trained model resnet-50 and three discriminators (SVM, KNN, and Naïve Bayes). The model is applied and tested on CASIA V2.0 dataset [14]. The result of this algorithm was not promising as the choice of resnet-50 was not good enough for the forgery problem. Resnet-50 construction is very complex and it needs a massive processing time for performing the process of both training and testing, and a big memory allocation which it is not accepted and valid in the actual forgery real problem-solving.

Qi, G. et al. [15] proposed a framework structure consisting of 15 layers (5 convolutional layers, 2 pooling layers, four layers RPN(regional proposal network), 1 ROI pooling layer, 2 fully connecting layers and 1 output layer). This model used max-out as an activation function in the convolution layers. The detection process was made using three stages: 1—ROI extraction by applying the maximum variance algorithm combined with morphological operations. 2—The 15 layers model was used to extract the dominant features. 3—Classification of the results to get the exact ROI. The strength of this model is using RPN in the designed model. RPN can efficiently and rapidly inspect locations to

determine if it required more handling in a certain area. The major problem this model faced was the first stage of ROI extraction applied. Firstly, an image was converted to grayscale and then applying the maximum variance and morphological operations. After this process, they reconverted the image again to color space. They lost a lot of details in the process of converting and reconverting from gray to color image. This process was considered to be one of the forgeries and editing applied to the image. This is the reason why the detection results were not satisfying enough. The recommendation to advance this model is to omit the first step of applying the maximum variance with morphology and applying batch normalization to their model. This will give a perfect result and can be applied in different applications.

As this paper is inspired by the AlexNet model architecture that was published and announced in 2012 [16], we searched and emphasized the study done on the previously published work that is based on the AlexNet model. It is precious to mention that there are three research papers, the ultimate found and known, which focalize their research on AlexNet specifically.

J. Ouyang et al. [17] proposed a framework that can only detect copy–move forgeries using AlexNet structure directly without any modifications to the network topology. They applied AlexNet on the ImageNet database. They applied AlexNet model on UCID, OXFORD flower, and CMFD datasets. The model obtained a good performance to the forgery image generated automatically by computer with a simple image copy–move operation, but is not robust to the copy–move forgery image of real scenario. The result was not satisfied enough and not robust to copy move in a real scenario. They also proved the concept that AlexNet can perform well in the forgery detection issue, and it was the first implementation of AlexNet in forgery detection. This work was the inspiration of other authors to start working on AlexNet as pre-trained network architecture.

A. Doegar et al. [18] proposed AlexNet model-based deep with SVM classifier to be applied to the available benchmark dataset MICC-F220. The training was done by training SVM using AlexNet as deep features and for testing, the test images are applied to the trained SVM to determine whether or not the test image is forged. This model structure yields great results for the MICC-F220 dataset as it consists of geometrical transformations of a genuine image's. The performance of the deep features extracted from the pre-trained AlexNet based model is quite satisfactory, the best accuracy of image forgery detection achieved is 93.94%. This proposed technique can only solve the problem of copy–move forgeries.

G. Muzaffer et al. [19] proposed a framework using AlexNet as a feature extractor and hence using the similarity measure between feature vectors to detect and locate the forgeries. They tested their technique on the available GRIP that includes copy–move forgeries [20]. This model was proven to give a more successful result on the GRIP dataset only. It was recommended to apply it on different datasets under different conditions.

Worthy massive research has been conducted on existing deep models for detecting and localizing digital image forgeries. The research investigates whether such techniques are sufficiently robust and whether they can properly model the manipulations that have occurred in images due to different types of forgeries that can faithfully classify an image as an authentic or fake image. This brief summary of the previously-published deep models clears that there is a high rising interest for novel solution models, to face the threats posed by increasingly sophisticated fake multimedia tools.

3. Proposed Work

The AlexNet model is nominated to be the solid core of the proposed model. The reason why we are using AlexNet, instead of any other pre-trained model is that we are planning to work with a simple model and test performances without compromising memory and time. Figure 2 shows the overall architecture of the proposed model; which is inspired by AlexNet but uses two different concepts. The structure of the proposed model is very similar to that of the original AlexNet model; which will be explained in Section 3.1 [21]. Both models have a similar number of layers, the same number of neurons, and the same-size filters. An improved framework is proposed by introducing

batch normalization and maxout as an activation function into AlexNet to resolve the drawback caused by AlexNet:

1. The obstacle of ReLU that can perish and never actuate on a single data point.
2. Modify the effect of normalization made by the local response normalization (LRN) exercised in the standard AlexNet. LRN is not trainable while batch normalization (BN) is trainable so the application of the later gives more promising results than LRN.

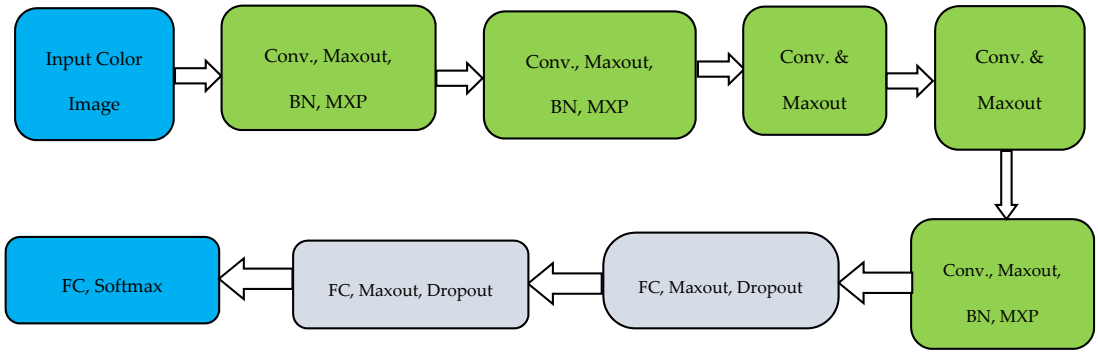

Figure 2. Overall architecture of the proposed AlexNet layers: convolution, max-pooling, max-out, BN and, FC.

The reason why AlexNet model repeated twice the layers (Conv., max-out, BN, MXP), (Conv. and max-out), and (FC, max-out, Dropout) is that the AlexNet was trained in a faster way by efficiently implementing the GPU of the convolution and all other processing in the training of CNN. AlexNet is therefore spread across two parallel GPUs which in turn fasten the processing speed of the model and take a smaller time to train the model.

3.1. The Proposed CNN Architecture

AlexNet model can yield high-performance accuracy measurements on different datasets. Whilst, detaching any of the convolutional layers must drastically decrease the AlexNet's effectiveness. The original AlexNet model network structure consists of eight consecutive layers, five convolution layers and three fully connected layers as shown in Figure 3 [22]. This deep structure of AlexNet leads it to be one of the best choices in the forgery process. All layers use a max-out activation function, excluding the last fully connected layer where the softmax function is applied. The core contributions are as follows:

1. Use max-out activation function instead of RELU for all the AlexNet layers.
2. Use batch normalization instead of LRN.
3. The proposed architecture will be used as a feature extractor for image input patches and as well as a classifier for the output result to detect the forgery result.

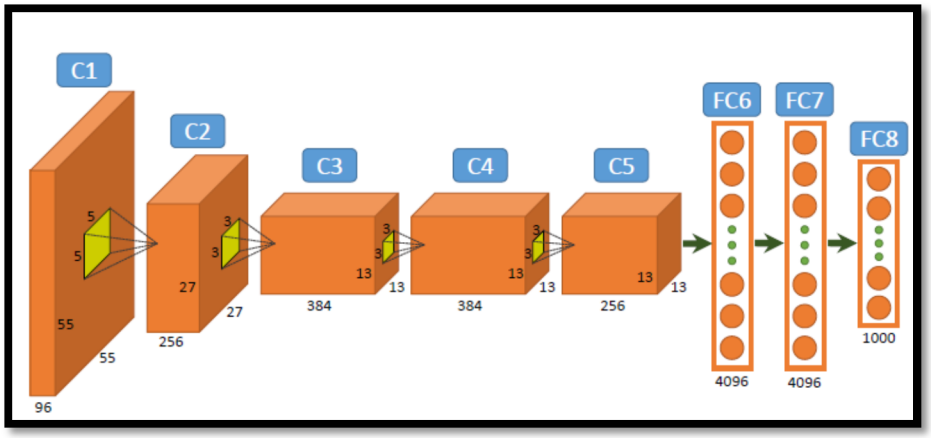

Figure 3. The overall architecture of AlexNet [23].

The input must be an RBG image of size 227 × 227. Without this image size, AlexNet suffers from considerable overfitting, which would have been forced to use much smaller network layers. If the input image is not RGB, it is modified to be an RGB image. If the input image's size is not, it will be converted to be of size 227 × 227. The first convolution layer performs convolution and max-pooling with BN where 96 different filters are used which are 11 × 11 in size. Consider an input image of size 227 × 227 × 3 that is applied to a convolution layer 1 with a square filter size 11 × 11 and 96 output maps (channels). Then layer 1 has:

- There are (227 × 227 × 96) output neurons in L, one per 227 × 227 "pixels" in the input and across the 96 output maps.
- There are (11 × 11) × (3 × 96) weights, (11 × 11 × 3) per filter (the input size run through the kernel), and 96 kernels in total (one for each output channel).
- There are (227 × 227 × 3 × 11 × 11 × 96) connections available. Single filter processes (11 × 11 × 3) values in the input; this occurs for each of the (227 × 227 × 96) output units.

This is similarly repeated through the next four convolution layers. Each layer has its own input size, filters size, the corresponding numbers of filters and output maps.

On the other hand, there are two fully connected layers, whichexercised with dropout succeeded by Softmax at the end of the model to act as the discriminant.

For getting families, understanding and explanation of the proposed work, a details will be elucidated in short and be focused on the significant terms used in the model:

1. Max Pooling (MXP): The proposed model utilizes a max-pooling technique that keeps only the maximum value in the filter to lower the dimension.
2. Dropout: This technique works as turning off nodes units with an agreed probability. We maintained a 50% dropout rate for the proposed AlexNet model. The reason for choosing a 50% dropout rate, it will give a maximum regularization of the model. That is because the dropout is used to minimize a loss function that follows a Bernoulli distribution [24].
3. Softmax Activation Function: An input vector x with p_i neurons are given, the softmax value of each neuron produces a corresponding output as in Equation (1).

$$y_j = \frac{e^{x_j}}{\sum_{k=1}^{k=K} e^{x_k}} \quad (1)$$

where x is the input vector to the output layer, j indicates the output units, so j = 1, 2... K, and K is the length of x.

4. Max-out Activation Function: The proposed model uses a max-out function as an activation function, instead of ReLU, since it is known to help fast convergence of large datasets. The max-out function [25] can be represented as follows in Equation (2).

$$max\left(w_1^T x + b_1, w_2^T x + b_2\right) \tag{2}$$

where: x is the input vector, w is the weight matrix and b is the bias. Max-out is well-known to be a learning activation function. ReLU is known to be a max-out special version. ReLU is a piecewise linear function that is easy to train and trivial to implement [26]. ReLU allows the model to be trained faster. Thus, the max-out activation function enjoys all the merits of a ReLU (operation linear regime, no saturation) and does not have its weaknesses (dying ReLU).

5. Batch Normalization is used for training a CNN that homogenizes inputs for each mini-batch. This has the impact of settling the learning procedure and dramatically minimizing the number of training epochs needed to train CNNs. BN has been used for the benefit of reducing Internal Covariate Shift (ICS) and accelerating the network training [27]. In BN, the output is handled in the following manner before going to the activation function:

   i. Normalize the whole batch B to be zero mean and one variance.
      - Calculate the mean of the entire batch output: μ_B
      - Calculate the variance of the entire batch output: σ_B^2
      - Normalize the batch by subtracting the mean and dividing by the variance.
   ii. Propose two training parameters (γ: for scaling and β: for shifting).
   iii. Apply the scaled and shifted normalized batch to the activation function.

Batch normalization normalizes inputs x_i through formulating μ_B and σ_B^2 for a mini-batch and input channel, after which it formulates the normalized activation as in Equation (3).

$$\hat{x_i} = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \tag{3}$$

where ϵ is applied to enhance the stability if the variance of the mini-batch is very small.

In the end of the network training, the BN hence develops both mean and variance across the whole training dataset, after which it retains them as properties named trained mean or trained variance. Compared to LRN, the LRN is a non-trainable layer that square-normalizes the pixel values in a feature map in a within a local neighborhood. LRN reduces activations that are uniformly huge for the neighborhoods which in turn creates a high contrast in a feature map. LRN is based on lateral inhibition which means performing a local maximum contrast [28]. BN has a regularization effect but LRN has not. Table 1 shows the differences between LRN and BN.

Table 1. Difference between local response normalization and batch normalization.

Normalization Type	Trainable	# of Trainable Parameters	Regularization
LRN	No	0	No
BN	Yes	2	Yes

4. Evaluation of the Proposed Work

This section discusses the details of the datasets used, the k-fold cross-validation, the experiment settings and environment, the performance evaluation policies, the experiments done and the comparisons of the results obtained. The experimental environment settings are explained in more details. For the proposed model performance measurement, diverse experiments evaluation have been

done and executed. Experimental results are represented to prove the proposed model's efficiency in detecting tampering and localizing it, and comparing performance with other related published work.

4.1. Datasets Description

For evaluating the proposed model performance, the used datasets were inspected, studying their performance, and then collating the proposed model to other key baseline models as a referral. Thus, we used CASIA v1.0, CASIA v2.0, NIST (National Institute of Standards and Technology) Nimble 2017, and DVMM [29] datasets for this purpose.

- NIST Nimble 17 dataset comprises around 10,000 images with numerous types of manipulations including the ones where anti-forensic algorithms were used to hide trivial manipulations. Ground-truth images' masks are in the hand for the evaluation process.
- CASIA v1.0, v2.0 datasets encompasses spliced and copy–moved tampered images altogether. The total number of images in CASIA v1.0 is 1721 (800 authentic and 921 spliced) and in CASIA v2.0 is 12,614 (7491 authentic and 5123 tampered images). They do not contain the ground truth masks, so ground-truth masks are obtained by thresholding the difference between tampered and original images.
- DVMM dataset encompasses only spliced tampered images. The total number of images is 1845 (912 forged and 933 original images). Ground truth images masks are at the hand for performance process evaluation.

4.2. K-Fold Cross-Validation

K-fold cross-validation emphasizes that the model has learned the dataset correctly [30]. The *k*-fold cross-validation method arbitrarily splits the dataset into equivalently sized enclosures, where *k* determines the number of partitions in which the dataset is split. The choice of an optimal *k* was often reported between 5 and 10, because the statistical performance did not raise that much for greater values of *k*, and averaging of less than 10 splits remains computationally feasible. The choice of *k* was a trade-off between the efficiency and the accuracy of the model. Multiple *k*-fold cross-validation techniques, like 5-fold, 8-fold, and 10-fold for examples, were applied to the best-fit training dataset of the proposed model, and we note that 10-fold is the best choice due to its lower sensitivity and less biased while separating data into training and testing. The choice of $k = 10$ depends on the training experiment and the accuracy of the model. There is no formal rule for choosing the number of *k*. If *k* was small, then the bias of the model to the dataset will be increased. Although a higher estimate of *K* decreased the bias, it may suffer from large variability. By applying $k = 10$; the dataset images are therefore partitioned into ten equal groups. Nine of these groups are counted to be the training dataset, while the one partition left was used for test data. Training was iterated ten times, every time using a diverse partition as a test group and the leftover nine partitions like training dataset. In the end, the mean result is considered as the final evaluation of the model.

4.3. Experiment Environment

To run the proposed model, all experiments are conducted on a machine with Intel(R) Core(TM) i7-5930K CPU @ 3.50GHz, NVidia GeForce GTX 2080 Ti with 16.0 GB memory in 64-bit window 8. The proposed model is implemented using anaconda navigator Python, Jupyter Notebook 6.0.2.

4.4. Performance Evaluation Policy

This sub-section pronounces the evaluation metrics used to evaluate performance. The evaluation metrics used are accuracy, precision, recall, and F1 Score. All of these evaluation metrics are derived from the four values that are listed in the confusion matrix as shown in Table 2 that is relied on the predicted class against the actual class [31]. True positive (TP) is defined as the forged images number that is detected as forged, true negative (TN) is defined as the number of the pristine images which

are detected as pristine, false positive (FP) is defined as pristine images numbers that are detected as forged and false negatives (FN) is defined as the number of forged images that are detected as pristine.

Table 2. Confusion matrix.

		Predicted Class	
		Malicious	**Benign**
Actual Class	**Malicious**	True Positive (TP)	False Negative (FN)
	Benign	False Positive (FP)	True Negative (TN)

1. Accuracy (acc) or Proportion Correct: A quotient of absolutely detected examples for all items. It is calculated as in Equation (4).

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

2. Positive Predictive Value (PPV) or Precision (p): A quotient of examples absolutely detected as X to all samples that were detected as X. It is calculated as in Equation (5).

$$Percision = \frac{TP}{TP + FP} \tag{5}$$

3. Sensitivity or True Positive Rate (TPR) or Probability of Detection (PD) or Recall (r): A quotient of examples absolutely detected as X to all examples that were exactly X. It is calculated as in Equation (6).

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

4. F1 Score (F1): The F1 Score is the subcontrary mean of precision and recall. It is calculated as in Equation (7).

$$F1 = \frac{2}{\frac{1}{r} + \frac{1}{p}} = \frac{2 * p * r}{p + r} \tag{7}$$

5. Matthews Correlation Coefficient (MCC): The MCC is normally used for evaluating the localization performance of each image in each dataset. The MCC is the cross-correlation between the model detection result and the ground-truth. It is calculated as in Equation (8).

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{8}$$

4.5. Experimental Results and Performance Evaluation

In this sub-section, we demonstrate the analysis and achievement of the proposed CNN model for image forgery detection. Comparing the performance is made between the proposed model and state-of-the-art similar models. The proposed model achieves very consistent performance across all testing datasets, indicating that it does generalize well on different datasets. Figure 4 displays different kinds of image forgeries manipulations that are done to the genuine images. Qualitative analysis of the proposed model results from different types of forgeries can be manifested as shown in Figures 5–8.

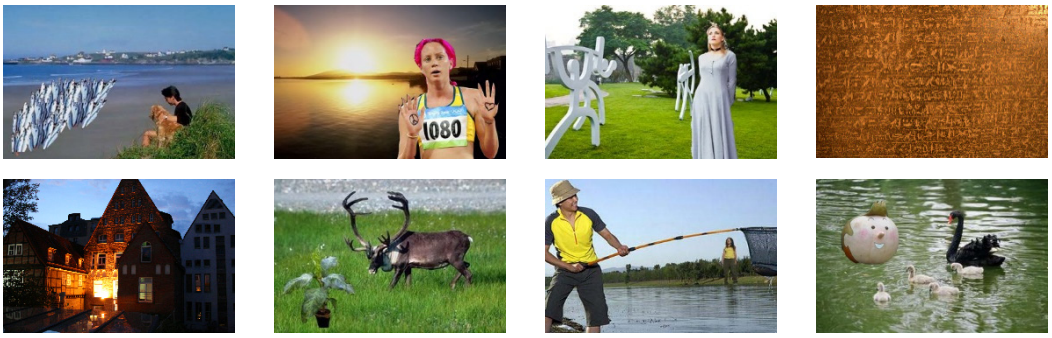

Figure 4. Examples of different kinds of image forgeries.

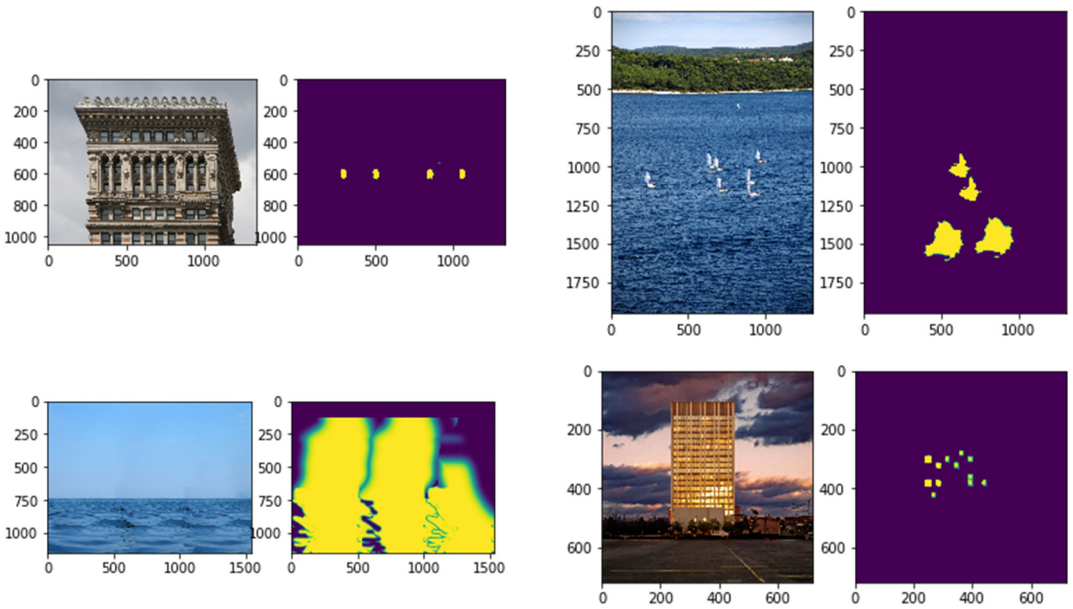

Figure 5. Some examples of multiple copy–move forgeries detection. The first and third columns represent the forged images. The second and fourth columns represent the detection results of the proposed model.

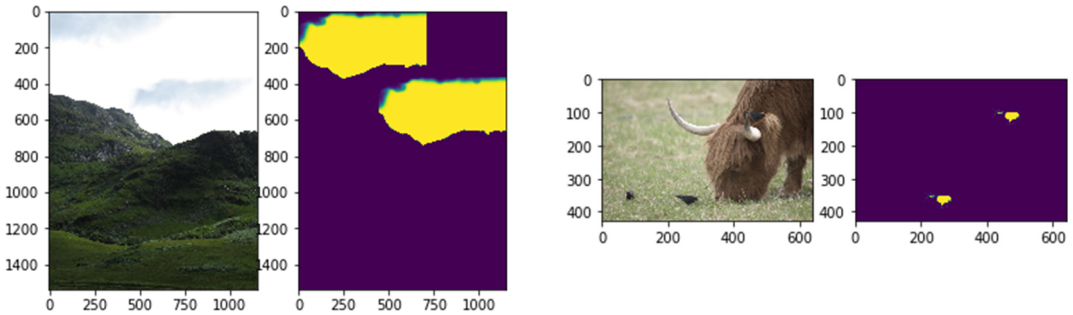

Figure 6. Some examples of forgery-dDetection. The first and third columns represent the forged images. The second and fourth columns represent the detection results of the proposed model.

Figure 7. Some examples of forgery-detection. The first and third columns represent the forged images. The second and fourth columns represent the detection results of the proposed model.

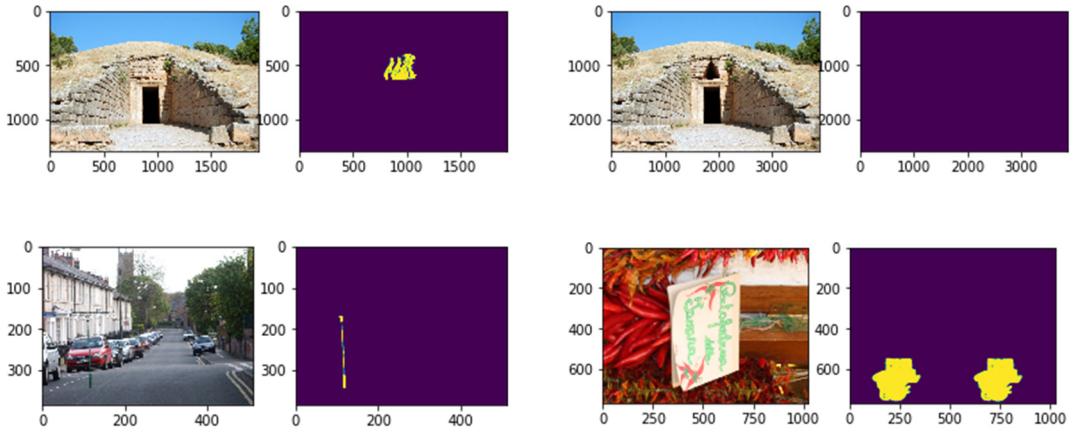

Figure 8. Different samples of forgery-detection. The first and third columns represent the forged images. The second and fourth columns represent the detection results of the proposed model.

From Figures 5–8, the first and third columns show the forged images using different types of manipulation. The second and fourth columns show the color-coded result of the forgery detection using the proposed model. Thus, one can easily identify forged areas and distinguish it from the surrounded genuine areas. The forged and the copied regions are marked with a yellow color, while the original areas are marked with a dark purple color.

Figure 5, for example, shows different examples of multiple copy–move forgeries. A certain object is copied and pasted at different times in the same image. This object can be scaled, rotated and shifted before being pasted. The proposed model can detect all the objects that have been copied along with the original one; to give an alarm that all of these objects are the same. Figure 7, for example, shows examples of how the proposed model detects a splicing forgery in given images. The spliced objects can be detected perfectly due to their appearance, their spatial extent and their geometrical structure, which are completely different from the neighbor objects and background. The same detection that has been occurred in Figure 6; Figure 8, the proposed model can detect the changes happened in an image. The proposed model has the ability to detect such changes in backgrounds, structures of objects, spatial extent of objects, contrast variations, and definitely the sudden variations of colors; that's why it is important to deal with color images to keep the color factor while analyzing images.

To evaluate the proposed model performance effectiveness, the upcoming experiments are performed and run:

1. Differentiation manifested among the proposed model and the models proposed in [17–19].
2. Datasets cross-validation applying to the proposed model and the models presented in [17–19].
3. Evaluating the proposed model by using the evaluation metrics in Equations (4)–(8). After many times experiments, the mean of all the results obtained is considered to be the final result as shown in Figures 9–13 and Tables 3–8.

Table 3. The detailed results of 10-fold cross-validation on CASIA V1.0 using the proposed model.

	Iteration 1	**Iteration 2**	**Iteration 3**	**Iteration 4**	**Iteration 5**	**Iteration 6**	**Iteration 7**	**Iteration 8**	**Iteration 9**	**Iteration 10**
#of training images	1549									
# of test images	172									
Accuracy	96.93	96.62	96.768	96.81	97.1	97.05	96.97	96.99	96.93	96.31
Precision	97.881	97.7	97.2	96.878	97.322	97.657	96.956	97.45	97.32	97.932
Recall	92.97	92.76	93.14	92.98	93.05	93.002	92.98	92.967	93.034	93.254
F1- Measure	95.363	95.1672	95.128	94.89	95.14	95.273	94.925	95.157	95.13	95.537
MCC	96.465	96.268	96.23	95.93	96.275	96.385	96.027	96.302	96.321	96.619

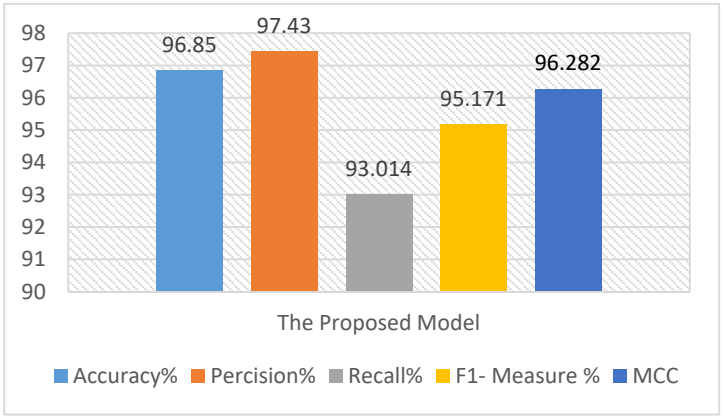

Figure 9. The 10-fold cross-validation average result on CASIA V1.0 using the proposed model.

Table 4. The detailed results of 10-fold cross-validation on CASIA V2.0 using the proposed model.

	Iteration 1	**Iteration 2**	**Iteration 3**	**Iteration 4**	**Iteration 5**	**Iteration 6**	**Iteration 7**	**Iteration 8**	**Iteration 9**	**Iteration 10**
#of training images	11,353									
# of test images	1261									
Accuracy	97.45	97.14	97.28	97.345	97.67	97.453	97.36	97.439	97.443	97.867
Precision	98.5	97.8	97.223	97.22	97.35	97.67	97.56	97.58	97.422	98.2
Recall	93.47	93.26	93.124	93.18	93.235	93.243	93.384	93.39	93.341	93.554
F1- Measure	95.92	95.4761	95.13	95.157	95.248	95.405	95.426	95.439	95.3385	95.8207
MCC	97.024	96.594	96.241	96.287	96.375	96.609	96.631	96.603	96.453	97.0321

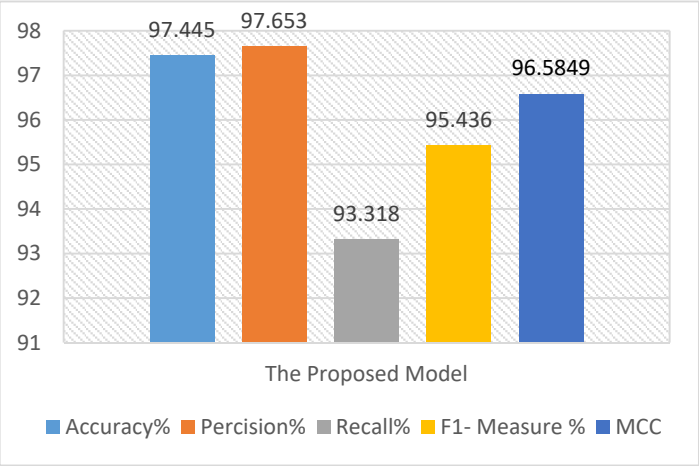

Figure 10. The 10-fold cross-validation average result on CASIA V2.0 using the proposed model.

Table 5. The detailed results of 10-fold cross-validation on DVMM using the proposed model.

	Iteration 1	**Iteration 2**	**Iteration 3**	**Iteration 4**	**Iteration 5**	**Iteration 6**	**Iteration 7**	**Iteration 8**	**Iteration 9**	**Iteration 10**
#of training images	1660									
# of test images	185									
Accuracy	97.01	96.943	96.868	96.92	97.176	97.125	96.97	97.09	97.33	97.52
Precision	97.951	97.743	97.2656	96.908	97.622	97.887	97.056	97.6	97.52	98
Recall	93.07	92.85	93.163	93.08	93.074	93.012	93.086	93.067	93.096	93.354
F1- Measure	95.45	95.236	95.171	94.926	95.295	95.388	95.03	95.279	95.2578	95.622
MCC	96.567	96.345	96.259	96.023	96.468	96.456	96.206	96.965	96.4695	96.895

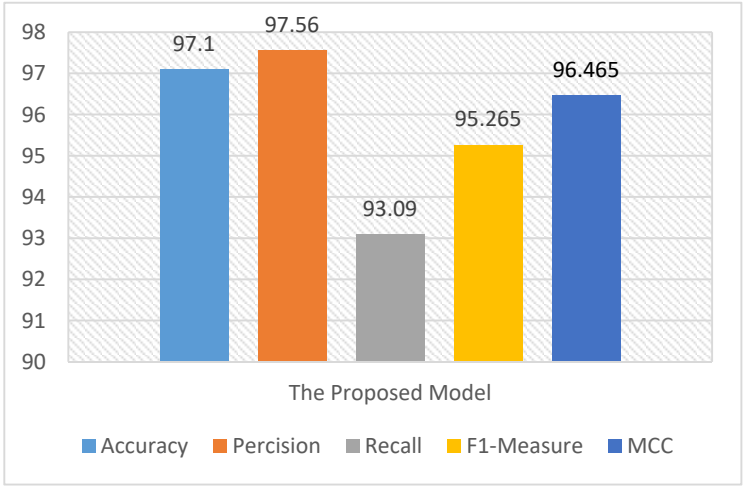

Figure 11. The 10-fold cross-validation average result on DVMM using the proposed model.

Table 6. The detailed results of 10-fold cross-validation on the NIST 17 dataset using the proposed model.

	Iteration 1	**Iteration 2**	**Iteration 3**	**Iteration 4**	**Iteration 5**	**Iteration 6**	**Iteration 7**	**Iteration 8**	**Iteration 9**	**Iteration 10**
#of training images	9000									
# of test images	1000									
Accuracy	97.33	97.02	97.16	97.33	97.54	97.43	97.248	97.36	97.39	97.75
Precision	98.52	97.83	97.54	97.22	98.12	97.67	97.56	97.58	97.422	98.423
Recall	93.35	93.14	93.05	93.004	93.12	93.154	93.234	93.123	93.278	93.323
F1- Measure	95.865	95.429	95.243	95.066	95.555	95.36	95.349	95.3	95.306	95.805
MCC	97.027	96.608	96.481	96.391	96.647	96.459	96.537	96.467	96.497	97.056

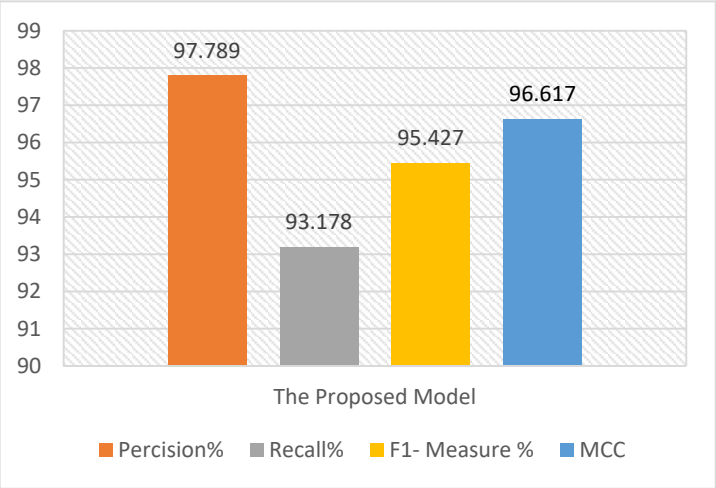

Figure 12. The 10-fold cross-validation average result on NIST 17 using the proposed model.

Table 7. The detailed results of 10-fold cross-validation on the overall datasets.

	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Iteration 6	Iteration 7	Iteration 8	Iteration 9	Iteration 10
#of training images	23562									
# of test images	2618									
Accuracy	98.1	97.64	97.85	98.5	98.19	98.053	98.643	98.47	97.943	98.367
Precision	98.78	98.32	97.93	97.952	97.85	98.38	98.544	98.08	97.72	98.346
Recall	94.17	93.66	93.924	93.98	93.835	94.045	94.503	93.79	93.911	94.254
F1- Measure	96.42	95.933	95.885	95.925	95.8	96.164	96.481	95.89	95.778	96.257
MCC	97.693	97.054	97.012	97.087	97.003	97.349	97.643	97.005	96.978	97.396

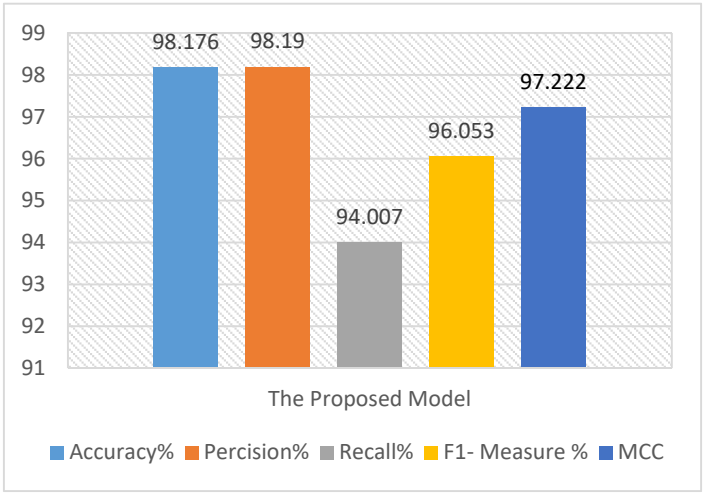

Figure 13. The 10-fold cross-validation average result applied on the overall datasets using our proposed model.

By deep scrutiny of Figures 9–13, it is remarkable that the proposed model gave a higher trigger response when the number of samples in the dataset increased. This is because the model is trained using a wide variety of samples which leads the model to be updated in order to detect different types of forgeries. For example, in Figure 13, as the model is trained using all the datasets mentioned, with a total number of training images 23,562, the model is well trained using a massive example. In this case, the proposed model gives a maximum value in terms of all evaluation metrics.

By careful study of Tables 3–7, it is noticeable that the evaluation metrics are varied by changing the number iteration on each dataset. Based on the k-fold cross validation concept and using 10 folds

for dividing datasets, the model splits each dataset into 10 groups. The model goes for 10 iterations, based on the numbers of folds applied, and swaps between groups in order to get nine groups as a training set and one group for the testing set. According to the Tables 3–7 and its recorded values, the model has the best values when it reaches the 10th iteration that is because datasets are varied in each group fold. This leads the model to be more generalized and capable of detecting different sorts of forgeries; which, in turn, gives a high score and promising results. If the number of folds increased by more than 10, there were no remarkable changes in the evaluation metrics and the results will be saturated.

Table 8. The overall average performance of the proposed model and other comparable models presented in [19–21].

	Accuracy	Precision	Recall	F1-Score	MCC
Model in [17]	96.231	97.32	93.001	95.111	96.732
Model in [18]	96.645	98.1	93.512	95.75	97.028
Model in [19]	95.856	98.056	93.453	95.699	96.978
Proposed Model	98.178	98.19	94.007	96.053	97.263

4.6. Comparative Performance Analysis

Having justified the proposed design choices and given a complete explanation of the proposed model, let us move to differentiate the proposed framework performance with those of comparable baselines, using diverse of datasets common in the image forgery detection issues. The results of the various experimental analyses of the proposed model using the modified version of Alex-net were compared with other forgery detection models using different structures of Alex-net, all in terms of accuracy, precision, recall, F1 score and MCC metrics. After describing and explaining the structure of the AlexNet layers and functions used, it is clear that AlexNet was a promising model to be used in the field of image forgery detection; that is, because of its deep and simple structure, its training speed, its less memory occupation and the solution of ReLU and LRN issues.

By deep scrutiny of the work done in this paper, it is apparent that the proposed model outperforms similar models using AlexNet model like models in [17–19]. This is because they all focus on the standard architecture of AlexNet, which only contains partial information for localization that limits their performance. The proposed model outperforms these models with the NIST17, CASIA v1.0, CASIA v2.0, and DVMM datasets. The proposed model captures global pixels rather than nearby pixels, which helps collect more cues such as contrast variation for the classification of manipulation. The 10-fold algorithm is utilized for dataset partitioning into training and testing, and examined the generalization capability of the model. Cross-validation is used to the utmost evaluation to reveal the weaknesses and assure the robustness of the image forgery detection model. By deep scrutiny of Tables 3–7, the evaluation metric values oscillate through 10 iterations of the cross-validation processing. The reason for this oscillation is that each run will permute data to generate a different dataset for training and another one dataset for testing. This is normal because the result's values are close and they do not vary that much. However, if the results metrics vary wildly, in this case using cross-validation is not valid for applying on the model. Table 8 summarizes the performance comparisons between the proposed models and similar models in [17–19]. The proposed model is ranked in the first place in the overall datasets used; this is possible because of the generalization ability of the proposed model.

By applying the cross-validation concept and constructing 10 different datasets, the model has been able to predict and work correctly on all datasets. When using cross-validation along with deep models, we ensure how accurate the proposed model is for many different datasets. We can, therefore, guarantee that the model generalizes perfectly to the dataset which will be applied later on. Consequently, we can say with confidence that cross-validation can improve the accuracy of the model and guarantee the generalization of the model.

Early deep learning architectures based on AlexNet, as models in [17–19], use a local response normalization layer which normalizes the central coefficient within a sliding window of a feature map considering its neighbors. Lately, Ioffe et al., presented in [32] the batch normalization layer that dramatically accelerates the training of deep networks. BN minimizes the internal covariate shift, which is a change in the inputs' distribution to a learning system. This has been performed by using the data zero-mean and unit-variance conversion whilst training the model. Each layer input has been influenced by the parameters of the preceding layers and even small changes get amplified. Thus, this type of layer addresses an important problem and increases the final accuracy of a CNN model. By using batch normalization, small changes in parameter to one layer do not get propagated to other layers. This makes it feasible to use greater learning rates for optimization. It also makes gradient propagation in the network more stable. Thus, using BN in AlexNet, as proposed in this work, instead of LRN, gives promising results in image forgeries using different datasets; which outperforms different proposed models using AlexNet in image forgery detection.

Comparing with the model proposed in [19], which used SVM as a classifier for the resultant values from the AlexNet instead of using its last fully connected layer with a softmax activation function. To differentiate SVM with Softmax, the SVM can be considered and classified as if it is a local objective [33]. So, SVM can intuitively be thought of as a feature. Softmax is highly used in the field of deep learning and gives a better classification output. So, softmax outperforms the result of SVM when applying to the problem of image forgeries. Thus, the proposed model gains greater and higher forgery detection results than the methods used in [18,19].

By deep investigation of the evaluation metrics used, it is clear that the proposed improved AlexNet model outperformed the previously published related work on different datasets. The model triggers higher scores using all performance measures used, which authorizes the model to be used in many forgeries problems.

5. Conclusions

In this paper, a modified deep CNN model based on a pre-trained AlexNet for image forgery detection and localization is proposed. The proposed work shows that the proposed model is deemed to be one of the best models to detect tampered images. Not only it is able to acquire performance much better than other models previously published, but it is also strongly robust to the most known image processing. Plainly, the proposed model can cope with a variety of operations with a strong learning capability. Inclusive experimental results presented that the proposed model is masterful in catching manipulations and attains good generalizability to unseen data and obscure editing types. The experimental results also show that the improved AlexNet model proposed for detecting and locating the forged areas score an effect that is better than the existing models on the datasets aforementioned. The detection results of people in different postures were also proved to be excellent. The improved AlexNet is proved to have the capability to learn the outlines of the forged areas and thus the capability to distinguish between the tampered and non-tampered areas. Even with promising results, it is a must to keep in mind that no model can solve all forgery attacks editing by itself. The model still needs further research to detect small forged areas and regions under massive variations.

6. Future Work

- Designing deep learning models to learn from smaller data: Deep learning models have been used for applications where huge amounts of unsupervised data are required. Deep learning has greater success with giant numbers of unlabeled training datasets. However, when the training dataset accessible is small, potent models are needed to gain improved learning capability. As a consequence, research on how to develop a deep model learning from the small training dataset is highly recommended.

- Applying optimization techniques to adjust the model's parameters: Adjusting the parameters in machine learning algorithms is an emerging topic in computer science. In deep learning CNN models, parameters that are needed to be adjusted is massive. Over and above, due to the hidden units' great number, the model is more probably gotten snared in the local peak optimal. Optimization techniques, e.g., PSO [34], are hence needed to solve this issue. The proposed model, therefore, should be capable of adjusting the parameters and extracting the features automatically.

Author Contributions: Conceptualization, S.S.; methodology, S.S., and E.E.; software, S.S.; validation, S.S., E.E., K.E.-S., and H.O.; investigation, S.S., E.E., K.E.-S., and H.O.; writing—original draft preparation, S.S.; supervision, E.E., K.E.-S., and H.O. All authors have read and agree to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare that there is no actual or potential conflict of interest regarding the publication of this article.

References

1. The 2017 Nimble Challenge Evaluation Datasets. Available online: https://www.nist.gov/itl/iad/mig/nimble-challenge-2017-evaluation (accessed on 28 September 2019).
2. Hadji, I.; Wildes, R.P. What do we understand about convolutional networks? *arXiv* **2018**, arXiv:1803.08834v1.
3. Rao, Y.; Ni, J. A deep learning approach to detection of splicing and copy-move forgeries in images. In Proceedings of the 2016 IEEE International Workshop on Information Forensics and Security (WIFS), Abu Dhabi, UAE, 4–7 December 2016; pp. 1–6. [CrossRef]
4. Liu, Y.; Guan, Q.; Zhao, X.; Cao, Y. Image Forgery Localization based on Multi-Scale Convolutional Neural Networks. In Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security—IH&MMSec '18, Innsbruck, Austria, 20–22 June 2018; pp. 85–90.
5. Bayar, B.; Stamm, M.C. 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, Vigo Galicia, Spain, 20–22 June 2016; pp. 5–10.
6. Cozzolino, D.; Poggi, G.; Verdoliva, L. Recasting Residual-based Local Descriptors as Convolutional Neural Networks. In Proceedings of the 5th ACM Workshop on Challenged Networks-CHANTS '10, Philadelphia, PA, USA, 20–21 June 2017; pp. 159–164.
7. Zhou, P.; Han, X.; Morariu, V.I.; Davis, L.S. Learning Rich Features for Image Manipulation Detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1053–1061.
8. Marra, F.; Gragnaniello, D.; Cozzolino, D.; Verdoliva, L. Detection of GAN-Generated Fake Images over Social Networks. In Proceedings of the 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), Miami, FL, USA, 10–12 April 2018; pp. 384–389.
9. Rossler, A.; Cozzolino, D.; Verdoliva, L.; Riess, C.; Thies, J.; Niessner, M. FaceForensics++: Learning to Detect Manipulated Facial Images. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
10. Salloum, R.; Ren, Y.; Kuo, C.-C.J. Image Splicing Localization using a Multi-task Fully Convolutional Network (MFCN). *J. Vis. Commun. Image Represent.* **2018**, *51*, 201–209. [CrossRef]
11. Zhang, Z.; Zhang, Y.; Zhou, Z.; Luo, J. Boundary-based Image Forgery Detection by Fast Shallow CNN. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 2658–2663.
12. Choi, H.-Y.; Jang, H.-U.; Kim, D.; Son, J.; Mun, S.-M.; Choi, S.; Lee, H.-K. Detecting composite image manipulation based on deep neural networks. In Proceedings of the 2017 International Conference on Systems, Signals and Image Processing (IWSSIP), Poznan, Poland, 22–24 May 2017; pp. 1–5.
13. Jaiswal, A.K.; Srivastava, R. Image Splicing Detection using Deep Residual Network. *SSRN Electron. J.* **2019**, *8*, 102. [CrossRef]
14. Dong, J.; Wang, W. CASIA v1.0 and CASIA v2.0 Image Splicing Dataset. Available online: https://www.kaggle.com/sophatvathana/casia-dataset (accessed on 28 September 2019).

15. Qi, G.; Wang, H.; Haner, M.; Weng, C.; Chen, S.; Zhu, Z. Convolutional Neural Network Based Detection and Judgment of Environmental Obstacle in Vehicle Operation. *CAAI Trans. Intell. Technol.* **2019**, *4*, 80–91. [CrossRef]
16. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Pdf ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
17. Ouyang, J.; Liu, Y.; Liao, M. Copy-move forgery detection based on deep learning. In Proceedings of the 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Shanghai, China, 14–16 October 2017; pp. 1–5. [CrossRef]
18. Doegara, A.; Duttaa, M.; Kumar, G. CNN based Image Forgery Detection using pre-trained AlexNet Model. *Proc. Int. Conf. Comput. Intell. IoT (ICCIIoT)* **2019**, *2*.
19. Muzaffer, G.; Ulutas, G. A new deep learning-based method to detection of copy-move forgery in digital images. In Proceedings of the 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT), Istanbul, Turkey, 24–26 April 2019. [CrossRef]
20. Cozzolino, D.; Poggi, G.; Verdoliva, L. Copy-move forgery detection based on PatchMatch. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 5312–5316.
21. A Walk Through AlexNet. Available online: https://medium.com/@smallfishbigsea/a-walk-through-of-alexnet-6cbd137a5637 (accessed on 5 April 2020).
22. Architecture of AlexNet. Available online: https://iq.opengenus.org/architecture-and-use-of-alexnet/ (accessed on 19 May 2020).
23. Available online: https://www.saagie.com/blog/object-detection-part1/ (accessed on 2 October 2019).
24. Understanding Dropout with the Simplified Math Behind It. Available online: https://towardsdatascience.com/simplified-math-behind-dropout-in-deep-learning-6d50f3f47275 (accessed on 5 May 2020).
25. Goodfellow, I.J.; Warde-Farley, D.; Mirza, M.; Courville, A.; Bengio, Y. Maxout Networks. *arXiv* **2013**, arXiv:1302.4389v4 stat.ML.
26. A Gentle Introduction to the Rectified Linear Unit (ReLU). Available online: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/ (accessed on 4 May 2020).
27. Romero, F.P.; Tang, A.; Kadoury, S. Multi-Level Batch Normalization in Deep Networks for Invasive Ductal Carcinoma Cell Discrimination in Histopathology Images. In Proceedings of the 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), Venice, Italy, 8–11 April 2019. [CrossRef]
28. What Is Local Response Normalization in Convolutional Neural Networks. Available online: https://prateekvjoshi.com/2016/04/05/what-is-local-response-normalization-in-convolutional-neural-networks/ (accessed on 4 May 2020).
29. Ng, T.T.; Chang, S.F. A Dataset of Authentic and Spliced Image Blocks Dept. Elect. Eng., Columbia Univ., New York, NY, USA, Tech. Rep. 203. Available online: http://www.ee.columbia.edu/ln/dvmm/newDownloads.htm (accessed on 27 September 2019).
30. Sebastian, R. Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. *arXiv* **2018**, arXiv:1811.12808v2 cs.LG.
31. Understanding Confusion Matrix. Available online: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62 (accessed on 5 October 2019).
32. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.
33. Qi, X.; Wang, T.; Liu, J. Comparison of Support Vector Machine and Softmax Classifiers in Computer. In Proceedings of the Second International Conference on Mechanical, Control and Computer Engineering (ICMCCE), Harbin, China, 8–10 December 2017; pp. 151–155.
34. Zeng, N.; Wang, Z.; Zhang, H.; Alsaadi, F.E. A Novel Switching Delayed PSO Algorithm for Estimating Unknown Parameters of Lateral Flow Immunoassay. *Cogn. Comput.* **2016**, *8*, 143–152. [CrossRef]