DETECTION OF COPY-MOVE FORGERY IN DIGITAL IMAGES USING DIFFERENT COMPUTER VISION APPROACHES

by

©Younis E. Abdalla

A thesis submitted to the School of Graduate Studies in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Faculty of Engineering and Applied Science

Memorial University of Newfoundland

March 2020

St. John's Newfoundland

Abstract

Image forgery detection approaches are many and varied, but they generally all serve the same objectives: detect and localize the forgery. Copy-move forgery detection (CMFD) is widely spread and must challenge approach. In this thesis, We first investigate the problems and the challenges of the existed algorithms to detect copy-move forgery in digital images and then we propose integrating multiple forensic strategies to overcome these problems and increase the efficiency of detecting and localizing forgery based on the same image input source. Test and evaluate our copy-move forgery detector algorithm presented the outcome that has been enhanced by various computer vision field techniques. Because digital image forgery is a growing problem due to the increase in readily-available technology that makes the process relatively easy for forgers, we propose strategies and applications based on the PatchMatch algorithm and deep neural network learning (DNN). We further focus on the convolutional neural network (CNN) architecture approach in a generative adversarial network (GAN) and transfer learning environment. The F-measure score (FM), recall, precision, accuracy, and efficiency are calculated in the proposed algorithms and compared with a selection of literature algorithms using the same evaluation function in order to make a fair evaluation. The FM score achieves 0.98, with an efficiency rate exceeding 90.5% in most cases of active and passive forgery detection tasks, indicating that the proposed methods are highly robust. The output results show the high

efficiency of detecting and localizing the forgery across different image formats for active and passive forgery detection. Therefore, the proposed methods in this research successfully overcome the main investigated issues in copy-move forgery detection as such: First, increase efficiency in copy-move forgery detection under a wide range of manipulation process to a copy-moved image. Second, detect and localized the copy-move forgery patches versus the pristine patches in the forged image. Finally, our experiments show the overall validation accuracy based on the proposed deep learning approach is 90%, according to the iteration limit. Further enhancement of the deep learning and learning transfer approach is recommended for future work.

Acknowledgements

This work is funded by the ministry of higher education of Libyan Government, which managed by the CBIE in Canada and the author acknowledged both of them for their support. I want to express my great gratitude to my thesis advisory Dr. Mohamed Tariq Iqbal and Dr. Mohamed Shehata, who make most of the help and reviews to make this work existed. I would also like to thank my supervisory committee members Dr. Minglun Gong for his kind support.

Words cannot express my thankfulness to the support and encouragement provided by my parents (My father Elmadany and my mother Amina), my lovely wife, and my sweet kids during my studies.

Many thanks are due to my brothers and sisters for their sweet wishes and kind support. My friends and all my family members who never stop encouraging me during my study, many thanks to you.

Table of Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

In this chapter, we will predefined the different abbreviations in Computer vision, machine learning and computer Science that we have used in this work. The following table lists down those abbreviations :-

Abbreviations			
$CMFD$	Copy-move Forgery Detection	MSE	Mean Squared Error
TPR	True Positive Rate	TNR	True Negative Rate
FNR	False Negative Rate	FPR	False Positive Rate
PPV	Positive Predictive Value	NPV	Negative Predictive Value
TPP	True Positive Predictive	$SIFT$	Scale-Invariant Feature Transform
$JPEG$	Joint Photographic Experts Group	$TIFF$	Tagged Image File Format
PNG	Portable Network Graphics	DIP	Digital Image Processing
SSE	Sum of Square Error	DCT	Discrete Cosine Transform
CNN	Convelutional Neural Network	DNN	Deep Neural Network
ROC	Receiver Operating Characteristic	NNF	Nearest Neighbor Field
$GANs$	Generative adversarial networks	SVM	Support vector machine
PCA	Principal component analysis	SGD	Stochastic sub-gradient descent
VGG	Visual geometry group	ROC	Receiver operating characteristic
AUC	The area under the curve	$HD5$	Hierarchical data format
$F1score$	A measure of a test's accuracy	BCE	Binary cross entropy
ORB	Oriented FAST and rotated BRIEF	$SURF$	Speed up robust feature
$cGAN$	Conditional generative adversarial network	$cNets$	Capsule network

Chapter 1

Introduction

Advanced technology is currently the go-to equipment used by forgers via computer graphics and digital image processing. In fact, the use of digital imagery to create forgeries is one of the biggest problems emerging from technology. However, experts working together with law enforcement are devising systems that employ advanced algorithms in order to ferret out the forgeries [1][2]. What may be surprising to those not working in the field is that very few digital documents today (especially those produced from medical, legal and government sources) are entirely free of some aspect of forgery. Detecting forgery algorithms is possible but depends almost entirely on the image source.

Nowadays, digital photographs and documents are easily changed to suit the purposes of the user, with copy-move being the most popular approach to forgery [3]. Copy-move is considered a type of passive forgery [4][5] if there is no authentic image copy for the forged image, and it is quite widespread. Figure 1.1 below shows different kinds of common forgeries [6]. One classic approach to digital image forgery is enhancing. This is the easiest approach and is also considered the least risky (i.e., it has the lowest repercussion if the forger is caught).

To counteract these forgeries, active and passive detection mechanisms have been developed [7]. In the active approach, digital watermarking or signatures are employed to make documentation more concise and genuine [4][6]. These mechanisms all serve the same objectives, with the concise documentation and robustness of the image processing field making the professional authenticated research works more obvious and thus helping to eliminate the fake works. Forgery detectors essentially share the same fundamentals for detecting forgeries – namely, the image information which is either information included or attached. For example, a color image filter that used to enhance the image, the used acquisition phase, or the camera lens characteristics. All this information can be detected professionally by using the photo-response non-uniformity noise sensor (PRNU). PRNU is a powerful algorithm for detecting copy-move forgery and is unique for each camera [4]. Other powerful algorithms can also detect copy-move forgery. These algorithms all have three main processes: feature extraction, matching, and post-processing at the pixel level to reduce false alarms. Additionally, scale and rotation invariant feature selection is also important for providing robustness. The best example powerful algorithm that achieved high efficiency of forgery detection is PatchMatch algorithm. The PatchMatching algorithm defeats is faster than most other algorithms and uses a matching approach that applies a dense approximation field matching. The PatchMatching offset field will implement more efficiency and smoothness in detecting copy-move forgery. In order to speed the matching of the offset points, the PatchMatching algorithm in our work runs the Denes-field to find the nearest neighbor field (NN), as follows:

$$delta(s) = argmin_{\phi:s+\phi\in\Omega.\phi\neq 0} D(f(s), f(s+\phi)) \tag{1.1}$$

$$NN \approxeq s^{'} = s + \delta(s) \tag{1.2}$$

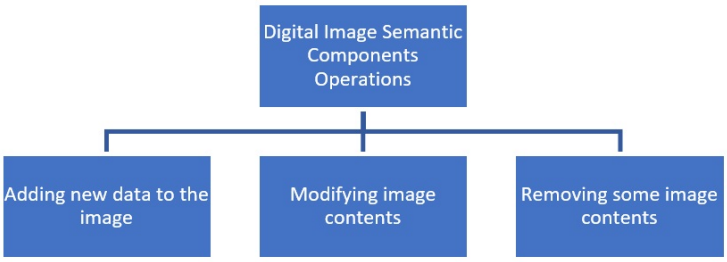

Figure 1.1: A Different types of forgeries

.

The new advances in deep learning, which has added a huge boost to this field, proposes of many new applications to support and leverage the computer vision technology, for instance, using Convolutional Neural Networks (CNNs). Nowadays, CNNs are most commonly used in computer vision applications and employ local neighborhood pooling operations and trainable filters when testing raw input images. Therefore, we propose novel models for copy-move forgery detection and localization using these new techniques.

The later chapters in this work will illustrate more details about our novel end-to-end network approach for image forgery detection and localization based on Scale Variant Convolutional Neural Networks (SVCNNs). After that, we will leverage this model by employing the Generative Adversarial Network (GAN) in dual branches of the model to achieve same objectives with more robustness.

1.0.1 Motivation

In recent years, our lives have become engrossed in the digital era, where technology dominates digital information usage, processing, and transformation. Images and their metadata have become easier to access and, as a result, software that makes it easier to tamper with and forge images has become more prevalent. This causes numerous problems regarding the authenticity of digital images, particularly in official processes. Examples include using images in the courtroom as evidence, in insurance claims for accident verifications, or in a medically-related way where an infection or other health anomaly can be edited-in and used as an official diagnosis for personal gain, like suing a workplace. There has also been an increase in the editing of personal photos for illegal gain, such as forging photo ID or various kinds of permits, legal documents, etc. Editing personal photos has even been used to damage people's reputation. The current state for used methods are suffering of low efficiency and/or lacking of specialization of the forgery type detection. These and other ongoing issues motivate us to explore this industry more in-depth in order to come up with a possible solution by building off state-of-the-art accomplishments, but with a focus on copy-move forgery.

1.0.2 Objectives of the Research

The main objectives of this thesis are due to:

1) integrate multiple forensic approaches to increase the efficiency of detecting and localize the copy-move forgery based on the same image input source.

2) improve the robustness of copy-move forgery detection models based on scale and rotation invariant features and metadata selection.

3) eliminate the dissimilarity feature-matching detection and increase the robustness of feature similarity detection in high dimensional feature space by employing the CNN model.

4) improve the computational speed implementation and operational cost of deep learning models for copy-move forgery detection by using the GAN algorithm and transfer learning technics.

1.1 Novelty and Research Contributions

The main contribution of this work is to enhance the capability of detecting and localizing copy-move forgery detection, by:

A) First, develop an algorithm to detect copy-move forgery detection that incorporates traditional methods that employ Patch-Match and dense-field algorithm. Other powerful algorithms can also detect copy-move forgery. These traditional algorithms all have the same three main processes: feature extraction, matching, and post-processing at the pixel level to reduce false alarms. Additionally, scale and rotation invariant feature selection is also important for providing robustness. The PatchMatching offset field increases efficiency and smoothness in detecting copy-move forgery. By running the Denes-field to find the nearest neighbor field (NN)to match the offset points in the forged seen main stone of enhancement. Applying this algorithm achieve a stronger efficiency of detecting and localizing copy-move forgery higher than 91%

B) Second, develop a learning algorithm based on CNN that can detect copy-move

forgery operations according to the feature similarity within an image frame. This novel scheme based on neural networks and deep learning, focuses on the convolutional neural network (CNN) architecture approach to enhance a copy-move forgery detection. The proposed approach employed a CNN architecture that incorporates pre-processing layers to gives satisfactory results. Additionally, the possibility of using this model for various copy-move forgery techniques is explained. The experiments show the overall validation accuracy is 90%, with a set iteration limit.

C) Third, distinguish the difference between the source patch and the target one in copy-move forgery detection by developing a learning algorithm that combines CNN and GAN in the end-to-end model. In this type of algorithm, the image is altered such that it appears identical to the original and is nearly undetectable to the unaided human eye as a forgery. The present approach investigated copy-move forgery detection using a fusion processing model comprising a deep convolutional model and an adversarial model. Four datasets were used. Our results indicated a significantly high detection accuracy performance (95%) exhibited by the deep learning CNN and discriminator forgery detectors. Consequently, an end-to-end trainable deep neural network approach to forgery detection appears to be the optimal strategy. The network was developed based on two-branch architecture and a fusion module. The two branches were used to localize and identify copy-move forgery regions, source patch and the target one, through CNN and GAN.

D) Finally, we used deep transfer learning for digital image forgery detection by using fewer resources to achieve higher efficiency forgery detection. The method applies prior knowledge that has been transferred to the new model from previous steganalysis models. Additionally, because CNN models generally perform badly when transferred

to other databases, transfer learning accomplished through knowledge transfer allows the model to be easily trained for other databases. The various models are then evaluated using image forgery techniques such as shearing, rotating, and scaling images. The experimental results, which show an image manipulation detection has validation accuracy of over 94.89%, indicate that the proposed transfer learning approach successfully accelerates CNN model convergence but does not improve image quality.

1.2 Thesis Outline

This thesis is written in manuscript format. The thesis is organized as follows:

Chapter 2 provides a literature review of forgery detection methods and the novelty and contributions of this thesis to the copy-move forgery detection. Chapter 3 focuses on a robust method for rotated and scaled copy-move forgery detection using enhanced PatchMatch algorithm. Chapter 4 presents a convolutional neural network for copy-move forgery detection. Chapter 5 illustrates copy-move forgery detection and localization using generative adversarial network and convolutional neural network. Chapter 6 presented an image forgery detection based on deep transfer learning. Finally, Chapter 7 concludes the presented work and predicts future work based upon the research and novel methods proposed.

1.3 References

[1] Mane, Vanita Agarwal and Vanita, "Reflection SIFT for Improving the Detection of Copy-Move Image Forgery," *ICRCICN*, Vol. 1, pp. 84-88, 2016.

[2] Anil Dada Warbhe, Rajiv V. Dharaskar, and Vilas M. Thakare, "Block Based Image Forgery Detection Techniques," *International journal of engineering science and research technology*, pp. 289-297, 1997.

[3] Gajanan K. Bitajdar, and Vijay H. Mankar, " Dgigtal image forgery detection using passive techniques: A survey," *Digital Investigation 10 (2013), Elsevier*, pp. 226-245, 2013.

[4] Al-Qershi OM, and Khoo BE, " Passive detection of copy-move forgery in digital images: state-of-the-art.," *Forensic Sci Int.*, pp. 1-3, 2013.

[5] Pravin Kakar, and N. Sudha, " Exposing Postprocessed Copy–Paste Forgeries Through Transform-Invariant Features," *IEEE Transactions on Information Forensics and Security*, vol 7, pp. 1018-1028, 2012.

[6] Rizvi, AB, " Digital Image Forgery Detection," Lincoln University, A New Zealand, 2015.

[7] Paweł Korus, " Digital image integrity – a survey of protection and verification techniques," Department of Telecommunications, AGH University of Science and Technology,*Elsever, Digital Signal Processing*, Vol 71, pp. 1-26, 2017.

Chapter 2

Literature Review of Forgery Detection Methods

2.1 Forgery Detection Methods' Overview

2.1.1 Introduction

In most cases, digital image forgery is simple image enhancement, but it can also be complete manipulation. Image retouching involves reconstructing damaged images, while image enhancement is an operation of shaping transformation. This operation has many positive advantages, such as, for example, restoring missing data [5]. However, it may also in some cases be illegal, especially when applied to official documents, academic work, or medical documentation. Image splicing involves copying and pasting more than one image into one component image, while image cloning is essentially another term for standard copy-move forgery.

2.1.2 Digital images forgery detection types

In recent years, developing a reliable method to detect digital image forgery is an active research area. Many techniques have been utilized for image forgery. The literature approaches can be simply classified into single and fusion approaches as the major sectors, as illustrated in Figure 2.1. On the other hand, image forgery detection mechanisms can be classified into two major categories of methods: active and passive. The active method can be presented either by digital fingerprinting, digital signature, or digital watermarking [1][2].

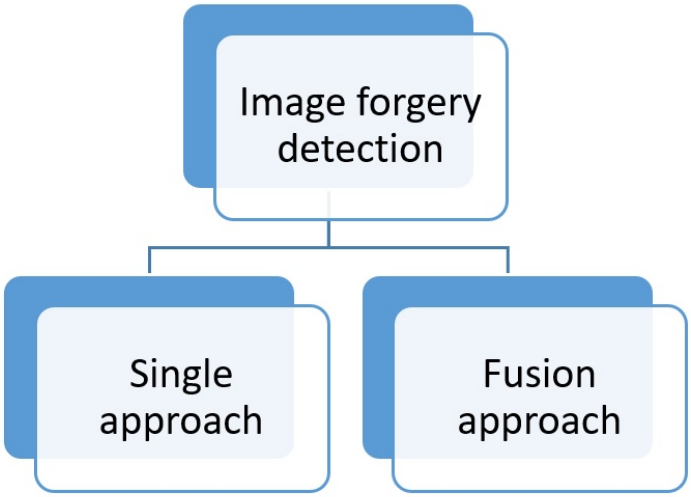

Figure 2.1: Digital image forgery approaches

.

2.1.3 Active forgery detection

An active forgery detection method, such as digital fingerprint, digital watermarking and/or digital signatures, uses a known authentication clue embedded or coded into

the image data before sending the image throughout an undependable transmission network. The algorithms used for this forgery detection will be briefly discussed next.

2.1.3.1 Digital fingerprint

This method determines the camera used in a forged image by extracting the sensor pattern noise. It employs a mathematical function to provide data, presenting a length value called a hash value or, simply, hashes. This length is a fixed step value among a several possible values: 32 bits, 64 bits, 128 bits, or 265 bits [3]. The digital fingerprint method shows high accuracy in identifying the source camera for many unknown images. However, it has limitations in that it tends to misclassify small hash values [4]. [5] the authors reported that the average accuracy of using fingerprint function is above 93%.

2.1.3.2 Digital signature

Digital signatures are obtained by concealing an image. The hash of the original image is taken and encrypted by RSA. The digital signature is sent along with the encrypted image which decreases the probability of a meticulous attack by the intruder. The encrypted image is shuffled using a Chaotic Logistic Map to obtain the final shuffled encrypted image. The use of the Logistic Map improves the randomness in the image. For authentication purposes, a comparator is employed, which evaluates the correctness of the hash extracted [6].

2.1.3.3 Digital watermarking

Watermarking is a technique for labeling digital pictures by hiding secret information within the images. Sophisticated watermark embedding is a potential method to discourage unauthorized copying or attest to the origin of the images [7].

2.1.4 Passive forgery detection

Digital image forgery can be found everywhere and is done for a variety of reasons. However, the major challenge is to determine if an image has been manipulated or not. Acquiring this knowledge solely with the aid of human eyes is impossible in most cases. On the other hand, applying forgery detection to every image is costly, time-consuming, and not always definitive. Detecting forgery based on a forged image is called passive forgery detection and is the most typical scenario in forgery activities. Passive or blind forgery detection technique uses the received images only for judging their truthfulness or authenticity, without any signature or watermark of the original image from the sender [8][55]. Therefore, passive forgery detection algorithms are better compared to active algorithms. The forgery detection process in this category is very unique and can be classified into two categories: Tampering forgery detection and source device forgery detection.

2.1.4.1 Tampering operations

In general, each algorithmic detector is designed based on how the forgery is made (e.g., copy-move and splicing are called dependent forgery). On the other hand, independent forgery can be classified into three ways: Compressing, Inconsistencies, and Re-sampling. All of these can be classified under tampering forgery. Therefore, the detector should first of all identify the tampering operation which was applied to the image and then makes the right decision regarding which algorithm needs to detect and localize the forged area within the targeted image.

2.1.4.2 Source device

This type of detection is used either as an optical or sensor regularities method to detect forgery. In source device identification, we try to identify the information about

the device, such as camera fingerprints, model, or other factory information.

2.2 Overview of the classic approaches used for copy-move forgery detection

2.2.1 Block Based Approach

This method is a fairly simple technique and yet quite effective in copy-move forgery detection. The technique maps the test image to small-sized patches or blocks (e.g., 4x4, 8x8 or 16x16) and uses these small patches in a specific sequence to scan all the images as a filter either horizontally, vertically or in a zigzag manner. By applying convolution between the filter and the same size small patches, the high convolution value result which exceeds the tested threshold will be defined as the duplicate area in the image. The best example of this technique is the Discrete Cosine Transform (DCT), as shown in Figure 2.2. Here, [9] divides the image to overlapping small blocks and then quantizes the DCT coefficients of each block, sorts them in the field of lexicography, and finds any similarities between these blocks. This approach is fairly robust to noise and image blurring but fails to detect copy-move forgery under geometric operations.

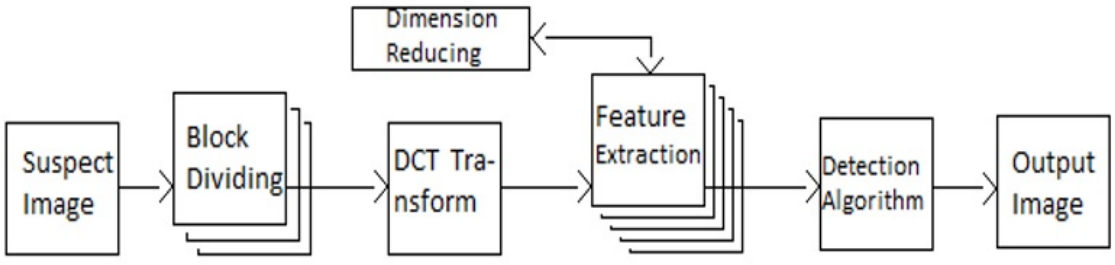

Figure 2.2: The flow chart of classic DCT method for copy-move forgery detection

.

2.2.2 Sensor Pattern Noise Based Approach

This method is used to prove the authenticity of an image based on sensor pattern noise or camera signatures. This can help to evaluate the truthfulness of an image by estimating the pattern noise of the same camera sensor. The noise pattern which is introduced by any type of camera will be divided into two main types: a random noise pattern and a fixed pattern noise. A random noise is changed from one exposure to another, whereas a fixed pattern is known as a Photo Response Non-Uniformity (PRNU). This is produced as a result of pixel light sensitivity and is highly noise-dependent. PRNU, which is a kind of intrinsic property of all digital cameras [10], is stable and unique for each camera. Therefore, by calculating the correlation between the designated image with the PRNU signal of the known camera, we can specify whether the image was captured by that camera or not. This will be obtained by using the PRNU pattern correlated with the query image noise residual and applying it at a specific threshold. If the correlation value is less than the threshold, the image was not captured by the known camera; otherwise, the image was captured by the camera. Figure 2.3 shows the algorithm used in the detection and localization forgery performance to estimate the accuracy and time conception to full processing. This is accomplished by measuring the F-measure. To do so, we need to determine all false-positive FP, true-positive TP, false-negative FN, and true-negative TN. The PRNU performs FM = 83% on average.

.

Figures 2.4 and 2.5 include the FM reading. When the detection map and ground

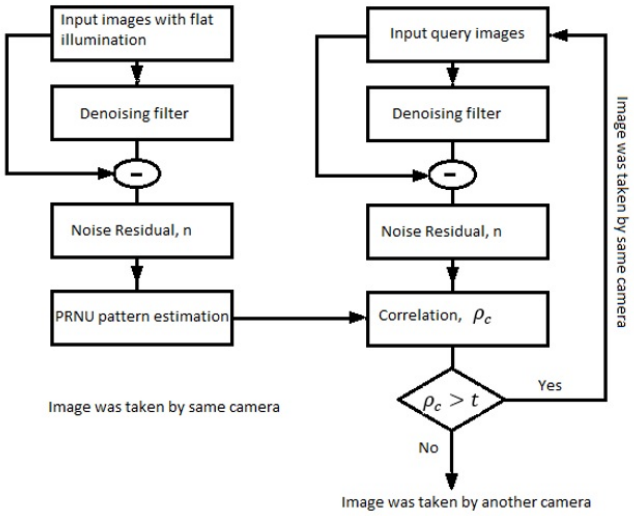

Figure 2.3: Shows The PRNU algorithm

truth occur at the same time or behave in the exact manner, then a false-negative FN and false-positive FP will equal to zero and the F-measure will be normalized (F = 1). The F-measure is obtained at two levels: the image level and the pixel level. To detect if there is forgery at the image level, the pixel level can be used to localize the forgery in the same image [11].

The accuracy of this approach depends on the true-positive rate (TPR) and the false-positive rate (FPR). The results show the PRNU approach has higher accuracy over the PatchMatch approach for detecting active copy-move forgery. In fact, the PRNU approach is specifically meant to deal with this type of forgery. We will show a comprehensive comparison of these two approaches in a later section.

.

.

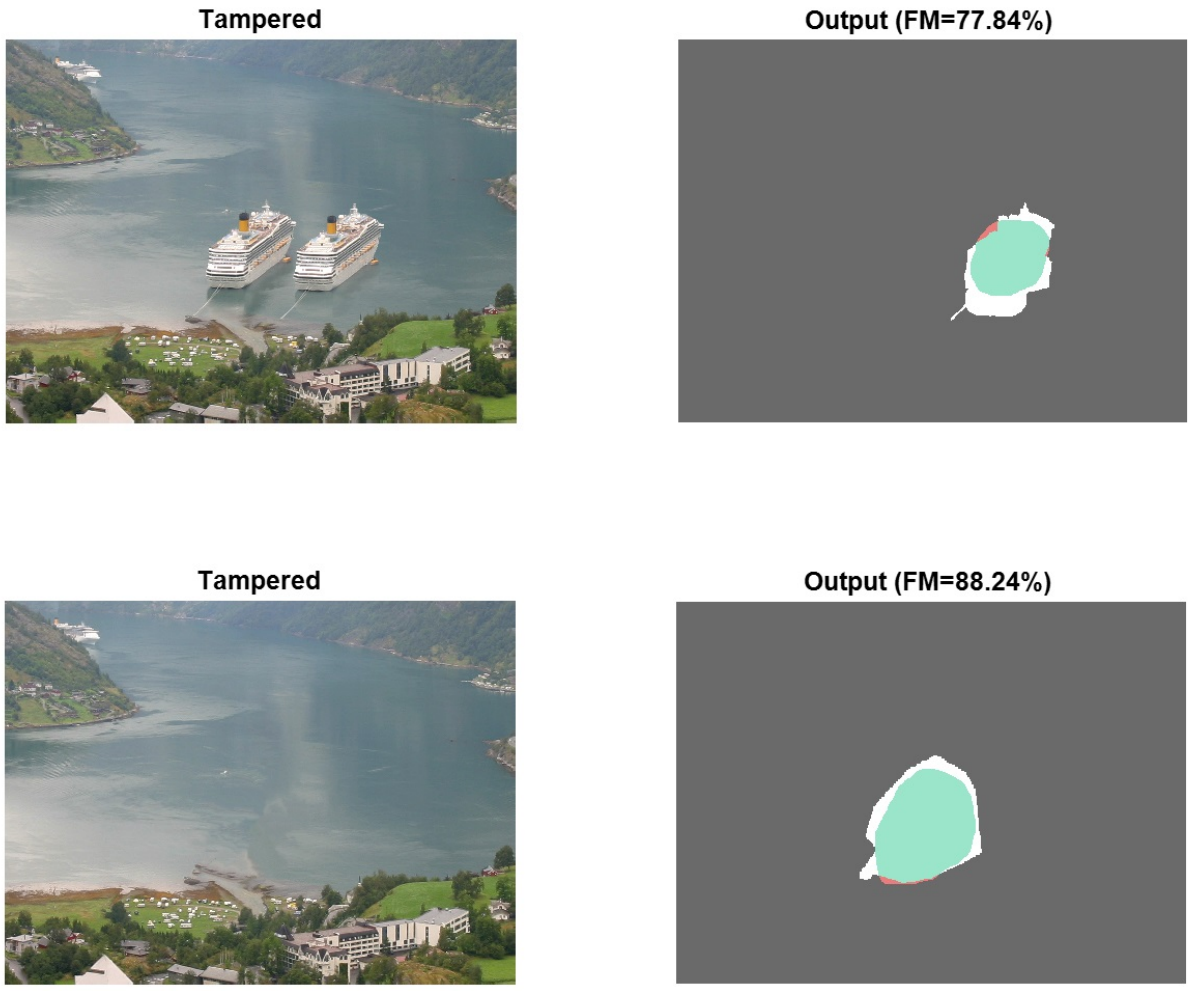

Figure 2.4: This figure shows a detection of copy-move forgery based on PRUN

Figure 2.5: This figure shows a copy-move forgery detection based on PRUN, the image from different camera

2.2.3 Copy-Move forgery detection using PatchMatch (dense)

A PatchMatching algorithm is faster than most other algorithms and uses a matching approach that applies a dense approximation field matching. The primary reason for choosing this approach over others is its fast propagation in offset fields. Iterations are performed by performing a randomized search or by doing full-image scanning (propagation). Generally, in a scan, we can first choose a specific vector f(s), that utilizes an s pixel for its patch center. Because the features essentially characterize the patch, the distance between and among the features needs to be carefully and accurately measured. The algorithm modifies the scaling, rotation and duplicate regions (copy-move) and finds a dense approximation of the nearest neighbor field (NNF) randomly, which matches the exact image's patches.

This algorithm will be discussed in detail in chapter 4.

2.3 Deep learning approaches for copy-move forgery detection

2.3.0.1 Method review

The high demand for technology requires much more processing in order to classify the origin of products. Digital image forgery and manipulation are largely intuitive, since smart tools make the task much easier and highly proficient. Researchers have applied different methods to detect forgery. In this study, as deep learning and neural network have been involved in wide applications and different purposes, the presented deep learning approach utilizes a convolutional neural network (CNN) to automatically detect forgery in digital images. The proposed method is applied for copy-move forgery detection based on convolutional neural networks (CNNs). The experiments demonstrate that the proposed CNN-based model with the preprocessing layers achieves excellent results.

In [12][13] and [14], deep learning methods applied to computer vision problems resulted in a local convolution feature data-driven CNN, while in other research, copy-move forgery detection algorithms were mostly based on computer vision tasks like image retrieval [15], classification [16], and object detection [17]. Along with CNNs, GPU technologies have helped to fuel the latest improvements in computer vision tasks [12]. Unlike traditional strategies for image classification, which mostly use local descriptors [18], the latest CNN-based image classification techniques use end-to-end structure. Because deep networks typically incorporate classifiers and features that are high, mid, or low level [19] using end-on-end multilayers, the various feature levels are enriched according to the numeral of hidden layers. The most recent convolutional neural networks like VGG [15], AlexNet [12], ResNet [21][14] and ResNet [22] significantly enhance performance in object detection and image

classification tasks [13]. Table 2.1 gives a summary of common CNNs [23], and shows some of the characteristics of these networks.

Table 2.1: The common CNNs characteristics

CNN	Layers No.	Year	Parameters No.	Error rate
LeNet	8	1988	60 T	N/A
AlexNet	7	2012	60 M	15.3%
ZFNet	7	2013	N/A	14.8%
GoogleNet	9	2014	4 M	6.67%
VGG Net	16	2014	140 M	3.6%
ResNet	152	2015	N/A	3.75%

.

2.3.1 Copy-move forgery detection based on convolutional neural network (CNN)

With the rapid development of a variety of digital image editing tools, editing images has become relatively easy. Editing of images also includes image forgery, which is commonly considered as a process of cropping and pasting regions on the same or separate sources [24].The goal of copy-move forgery detection is to determine the authenticity of an image by detecting the traces left by copy-move forgery. Copy-move forgery detection is one of the most actively investigated topics in image forensics. In general, there are two main branches in copy-move forgery detection: block-based forgery detection, and keypoint-based forgery detection [25]. In the block-based copy-move forgery detection methods, overlapping image patches that contain raw or transformed pixels are extracted, and similar patches are sorted to seek traces of forgery

[26]. There are numerous methods currently being used in image forgery detection. Many of these approaches consider the statistical characteristic in different domains [27]. However, image forgery localization is one of the most challenging tasks in digital image forensics. Different from forgery detection, which simply discriminates whether a given image is pristine or fake, image forgery localization attempts to detect the tampered areas [28]. In contrast to the aforementioned methods, we do not focus on any specific feature or any domain but concentrate on constructing a model to extract useful features automatically. Our model is based on a deep neural network. This type of network includes Deep Belief Network [29], Deep Auto Encoder [30] and Convolutional Neural Network (CNN) [31]. CNN is a widely used deep neural network model for computer vision applications. It applies trainable filters and local neighborhood pooling operations on the raw input image, resulting in a hierarchy of increasingly abstract features. CNNs have a high impact on computer vision and image understanding in general, so they can achieve superior performance on a visual object recognition [32]. This is because the structure and working principle of CNNs are highly similar to those of the visual system [33]. In fact, CNNs can be obtained simply by composing linear and non-linear filtering operations such as convolution and rectification [34].

2.3.2 Neural network architecture

The CNN architecture in this research has been built to perform forgery detection. As such, CNN is layered in a specific arranged and sequence and operates as a feature extraction which consists of a set of filters with a designated size. These filters convolve in parallel with all regions of an input image with an overlapping distance called a stride. The output of each convolutional filter in a convolutional layer is a newly learned representation of data known as the feature map. Subsequently, the following

hidden convolutional layers similarly extract features from the input feature maps previously learned by a former convolutional layer. The output of these hierarchical feature extractors is fed to a fully-connected layer that performs a classification task using learning weights [35]. These weights are initialed randomly, then learned by the backpropagation technique. The set of hierarchical convolutional layers yields a large volume of feature maps which makes the CNNs computationally very expensive [36]. The used network in this work has 15 layers: 3 convolution layers, one max-pooling layer, 2 average-pooling layers, 4 ReLU layers, 2 fully-connected layers, SoftMax layer, and the input and classification output layers.

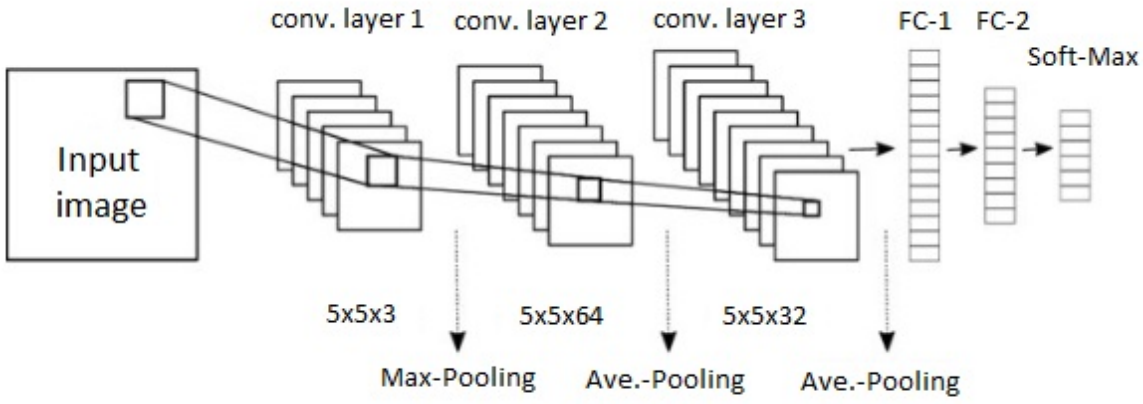

Figure 2.6: A Network architecture of the CNN model

.

2.3.3 CNN model identification

Convolutional Neural Networks is a complex nonlinear interconnection of neurons inspired by the biology of the human visual system. Successfully used in object recognition, face identification, image segmentation, etc., the use of CNNs in forensics

is relatively recent. Inspired by results obtained by [37],we take the proposed CNN as a feature extractor for each patch of the input image. The structure of a CNN is divided into several blocks, called layers. Each layer L_i takes as input either as

$$H_i * W_i * P_i$$

feature map or a feature vector of size P_i and produces as output either as

$$H_{(i+1)} \times W_{(i+1)} \times P_{(i+1)}$$

feature map or a feature vector of size $P_{(i+1)}$. Layer types used in this work are: Convolutional layer Performs convolution, with stride S_h and S_w along first two axes, between input feature maps and a set of $P_{(i+1)}$ filters with size $K_h * K_w * P_i$. Output feature maps have size of 5x5x32.

2.3.4 Copy-move forgery detection based on Siamese network

.

In order to enhance the performance of the network, we decided to train the features instead of training the image. To accomplish this, we used different networks to train the features for each category individually and match both outputs trained features to detect the copy-move based on the threshold convolution value or similarity vs. dissimilarity. A novel fully-convolutional Siamese network trained end-to-end is the best option to achieve this target at this time. Siamese networks are a special type of neural network architecture. They were first time introduced in 1990 by Bromley

and LeCun to solve the problem of signature verification based on image matching technique [38]. Instead of a model learning to classify its inputs, the neural networks learn to differentiate between or match two inputs. Specifically, they learn how to distinguish similarities between them.

Let the f_1, f_2 be a pair of input images to our neural network. Let α be a binary label. If the images have the same features, α will be labeled to 1, and the d function will calculate how much the images are similar. Otherwise, α will be labeled to 0 to present the difference between the pair input images. The d function is called a loss function. Through this function, we marry the two networks in one Siamese network and find the match between the two inputs in these networks.

$$d(f_1, f_2) = \sum_i \alpha_i |f_1[i] - f_2[i]|$$

2.3.4.1 Siamese architecture

The Siamese networks consist of two branches which present as two identical neural networks. Each branch takes one category of the two input images. The weight parameters between the twin network branches are tied, which guarantees that two exact similar images will have no possibility to miss their similar feature map during the detection process because they are dealing with the same functions. The last layers of the two neural networks are then fed to a one contrastive loss function, which detects the forgery or pristine in the tested image based on the similarity and/or dissimilarity between the two images. The two identical neural networks have some weights. Each pair of the images out of two different datasets presenting different categories based on the training propose which in this case copy-move forgery detection. Figure 2.7 shows the main structure.

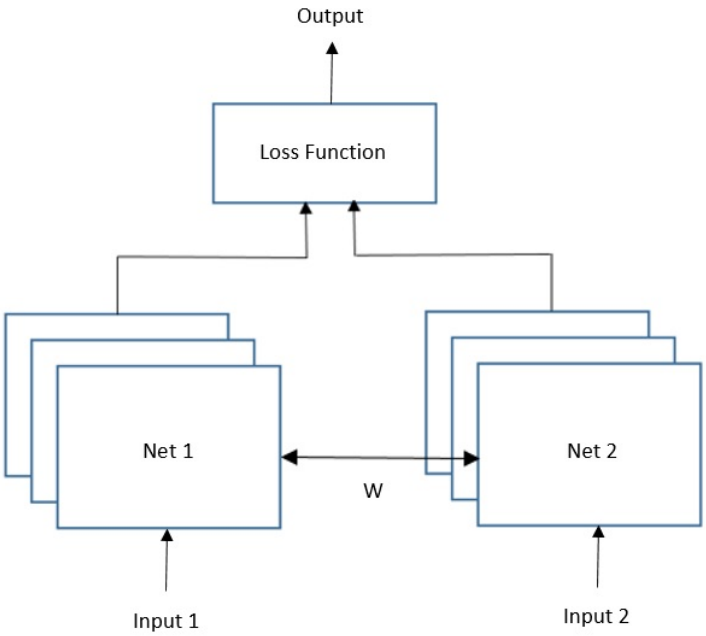

Figure 2.7: Diagram of Siamese network structure

By applying Siamese, three main aspects emerge, as follows:

i. The loss function in Siamese networks is very important to show similarity or dissimilarity.

ii. The architecture of Siamese networks is symmetric, which ensures the same processing to the two inputs.

iii. Mixed Network Architecture in one network can improve the performance of that network outcome.

The problem with copy-move forgery detection is localizing the copy moved area from the original one. Also, many forged images have no pristine image pairs, in which case the typical Siamese network will be a similarity detection tool only.

Deep convolution learning algorithms are one such solution. These are highly ef-

fective in dealing with image forgery derived from generative adversarial networks (GANs) [39]. In this type of algorithm, the image is altered such that it appears identical to the original image and is nearly undetectable to the unaided human eye as a forgery. In this work, a novel idea by employing the same Siamese network's structure by pairing to different networks CNN and GAN. CNN network to detect the similarity between any similar patches in the same image frame, while the GAN network, on the other hand, will detect the interrupted area [40][41]. If the image has copy-move forgery, the model will produce three outputs categories: the forgery patch mask, the similarity detection mask, and the original versus the copy-moved area mask.

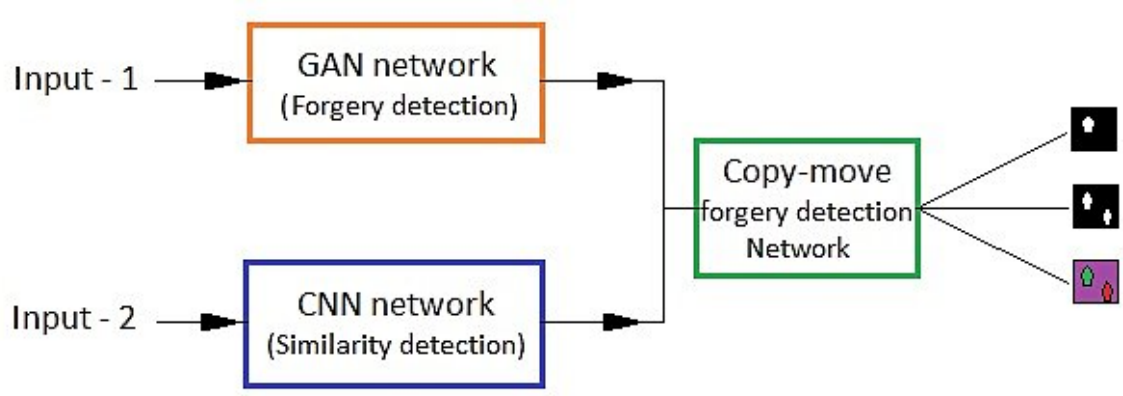

Figure 2.8: CNN-GAN model for copy-move forgery detection based on Siamese structure

Figure 2.8 illustrates the based diagram of the CNN-GAN network structure for copy-move forgery detection. The two branches are used to localize and identify copy-move forgery regions through CNN and GAN.

2.3.5 Forgery detection based on deep transfer learning

The main concepts of transfer learning as a means to reduce both the amount of resources required and the time it takes to train the networks especially when the used dataset is huge. Doersch et al. in [42], studied the behavior of learning features in unsupervised datasets. Their aim was to see whether or not a network is able to learn objects occurring within images; if a network can do this, it indicates an ability to learn underlying feature orientation for images. Therefore, by applying these learned representations, features learned from this network could potentially be reused in unsupervised object detection for other datasets, which can be translated as a type of transfer learning.

The key here is to reuse the pretrained network to do a new task when the cases have some common layout and to adjust the new case to fit with pretrained size similarity matrix. It is then necessary to fine-tune the new model based on the chosen pretrained model's matrix. Figures 2.9 and 2.10 provide a summary of the cases and the process, showing when and how the transfer learning process can be applied.

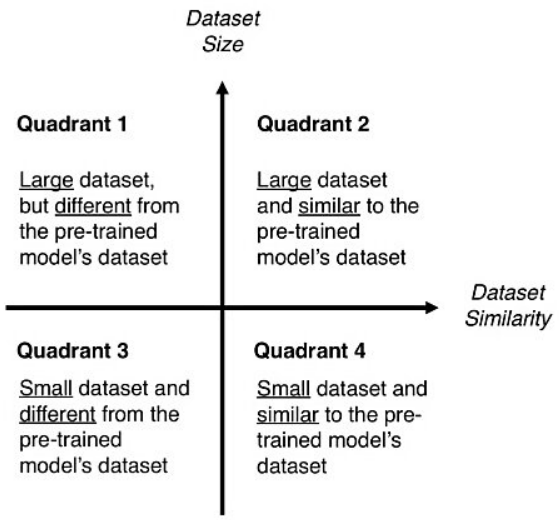

Figure 2.9: Size similarity matrix for pretrained model's datasets [43]

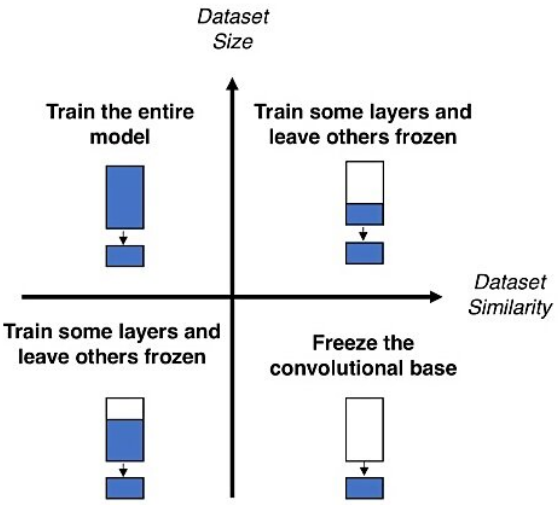

Figure 2.10: Decision map for fine tuning pretrained model [43]

2.4 Related Work

2.4.1 Introduction

The history of forgery is as old as mankind. Throughout the centuries, it has primarily been used as a means to acquire access to power or money illegally [44]. Although this motivation persists, many cases of forgery today are focused instead on gaining access to systems for a variety of purposes. So, for instance, people engage in forgeries across fields as diverse as healthcare, surveillance, insurance, and even the media. To counteract forging activities, researchers are exploring algorithms as a means to detect image forgery. In the majority of the algorithms used thus far, lighting is analyzed to see whether or not copy-move forgery is present. During the forgery process, the image becomes "messy", and it is this "mess" that forgery detectors look for through the application of algorithms, as explained in [45]. The researchers [45] also demonstrate how shadows can generate similar lighting artifacts within an image. As touched on earlier, there are several different algorithm-based approaches for forgery

detection, but the most popular techniques are block-based and feature-based. For block-based approaches, the detector needs access to the original image, whereas, for feature-based strategies, the detector removes features by means of overlapping blocks that are typically used in the block-based approach. Several various kinds of characteristics can be input to the overlapping blocks, and the matching among the boxes is performed on the basis of the feature-extraction strategy. In summary, this chapter aims to provide a comprehensive overview of the state-of-the-art in the digital image forgery. This technology has been invented and designed to determine whether the image content is authentic or manipulated, without any prior knowledge about the image. Also, it is used to distinguish between the source image and the target image.

2.5 Feature Types

Firstly, we overviewed different types of image features and then we included three types based features extractions: Polar Cosine Transform (PCT), Zernike Moments (ZM), and Fourier Mellin Transform (FMT). For the first two types, will be having two distinct categories: polar and cartesian.

2.5.1 Polar Cosine Transform (PCT)

Polar cosine transforms (PCT) is a fast algorithm which suits more for large images and real-time application, and it was proposed to represent the pattern of 2-D image $f(x, y)$ by transforming it from cartesian to polar form $f(r, \theta)$, where r is the reduce and θ is the azimuth.

$$r = \sqrt{x^2 + y^2} \tag{2.1}$$

$$\theta = arctan\frac{y}{x} \tag{2.2}$$

The polar form will be found as:

$$f(r,\theta) = \sum_{n=1}^{(\infty} \sum_{l=1}^{(\infty} M_{nl}^{c} H_{nl}^{c}(r,\theta) \tag{2.3}$$

Where $r \leqq 1$.

$$M_{nl}^{c} = \Omega_n \int_0^{2\pi} \int_0^1 f(r,\theta) H_{nl}^{c}(r,\theta) \rho d\rho d\theta \tag{2.4}$$

$$H_{nl}^{c}(r,\theta) = R_{nl}^{c}(r) e^{iln} \tag{2.5}$$

$$R_{nl}^{c}(r) = cos(\pi n r^2) \tag{2.6}$$

$$Omega_n = \{ \begin{matrix} \frac{1}{\pi} & \text{if} \quad n{=}0 \\ \frac{2}{\pi} & \text{if} \quad n{\neq}0 \end{matrix} \tag{2.7}$$

The PCT will be defined on the unit circle, and to generate the Kernel coefficient for each point, three trigonometric functions [46].

2.5.2 Zernike Moments Transformation (ZM)

Zernike moments are used for image recognition and find an image orientation, size, and position. So, it is basically an extinction of geometric moments and [47] describe the relationship between them. The Zernike function can be presented as follows:

$$V_{nm}(\rho,\theta) = R_{nm}(\rho)e^{jm\theta} for \rho \leq 1 \tag{2.8}$$

Where n,m are the order and the rotation respectively. $R_{nm}(\rho)$ is the radial polynomial, and it can be given as:

$$R_{nm}(\rho) = \sum_{x=0}^{(n-|m|)/2} (-1)^x \frac{(n-x)!}{x!(\frac{n+|m|}{2}-x)!(\frac{n-|m|}{2})} \tag{2.9}$$

The tow dimensional ZM for continuous image function $f(\rho,\theta)$ can be described as:

$$Z_{nm} = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(r,\theta) V_{nm}^*(\rho,\theta) \rho d\rho d\theta \tag{2.10}$$

$$Z_{nm} = \frac{n+1}{\pi} \int_0^{2\pi} e^{-jm\theta} \int_0^1 f(r,\theta) R_{nm}(\rho) \rho d\rho d\theta \tag{2.11}$$

In the digital image form in 2-D the ZM will be as:

$$Z_{nm} = \frac{n+1}{2} \sum_{(\rho,\theta)\in unitdisk} \sum f(r,\theta) V_{nm}^*(\rho,\theta) \tag{2.12}$$

The Zernike moment is rotation invariant, this helps to detect the rotated forgery. Therefore, the literature shows many algorithms use the Zernike moment to detect the forgery [48][49][50].

2.5.3 Fourier-Mellin Transform based feature extraction

The recent efficient block-matching based copy-move forgery detection approaches are using Fourier-Mellin Transform (FMT) which was proposed by [51]. In fact, this method performs radial projection on the log-polar organize Fourier transformation

of image blocks as follows:

a- Obtain translation invariant for each block i(x,y) by applying Fourier transformation-representation.

$$|I^{'}(f_x, f_y)| = |\sigma|^{-2}|I^{'}(\sigma^{-1}((f_x cos\alpha, f_y sin\alpha), (-f_x sin\alpha, f_y cos\alpha))| \tag{2.13}$$

b- Resample the magnitude values result into log-polar coordinates.

$$|I^{'}(\rho, \theta)| = |\sigma|^{-2}|I(rho - log\sigma), \theta - \alpha| \tag{2.14}$$

c- Project log-polar values onto 1-D, and obtain $\theta = 45$ features by quantization these summed values for different θ.

$$g(\theta) = \sum_i log(|I(\rho_j, \theta|) \tag{2.15}$$

The FMT achieve high performance in forgery detection of flat regions.

2.6 Feature Extraction

2.6.1 Feature Extraction based on Classic Approach

As mentioned previously, there are several kinds of features extraction methods available in published work and online. They all suggest the efficacy of one or more approaches for the detection of copy-move forgery, but this present work only looks into 3 main classifications of features, namely: the Fourier-Mellin transform (FMT), the Zernike moments (ZM), and the polar cosine transforms (PCT). The features

in all 3 of these approaches share highly similar circular harmonic transform expansions (CHT) [52]. Therefore, measuring the CHT coefficient through image projection $I(\rho, \theta)$ using the basis function of $K_(n, m)(\rho, \theta)$ to effect the transformation:

$$F_l(n, m) = \int_0^{\infty} R^*_{n,m}(\rho) \times [\frac{1}{\sqrt{2\pi}} \int_0^{2\pi} I(\rho, \theta) e^{-jm\theta} d\theta] d\rho \tag{2.16}$$

As can be seen, the image $I(\rho, \theta)$ appears in the polar scheme, with $\rho \in [0, \infty], \theta \in [0, 2\pi]$. This particular approach entails combining aspects of 2 formulations: 1. integrating the Zernike radial, function, and the ρ value integration; and, in brackets, indicating the Fourier series function for the image $I(\rho, \theta)$ together with phase term $e^{-jm\theta}$ with a rotation of θ radians. Thus, to obtain rotation invariance, we simply use coefficient magnitude, such that FMT's coefficient absolute value will then give scale invariance, as any alterations in image scale add to the phase term [53]. Hence, the radial function is then variant-based according to feature designation. Therefore, we can assert that PCT radial function acts as a cosine function arguing ρ^2 while normalizing coefficients C_n.

$$R_n(\rho) = C_n Cos(n\pi\rho^2) \tag{2.17}$$

In this case, the Zernike radial function demonstrates the identical radial function of PCT, but includes coefficient values that are more apt and uses the formulation $\rho \in [0, 1]$ in both functions. This is formulated as follows:

$$R_{n,m}(\rho) = \sum_{h=0}^{(n-|m|)/2} C_{n,m,h} \rho^{(2-2h)} \tag{2.18}$$

At the same time, we can show the FMT radial function as non-zero in $\rho \geq 0$, using a continuous value r above the argument value of ρ^2

$$R_(\rho) = (\frac{1}{\rho^2} e^(jrln(\rho)) \tag{2.19}$$

2.6.2 Feature Extraction Using CNNs

Feature extraction in neural network shows how to extract learned image features from a pretrained convolutional neural network and to use those features to train an image classifier. Feature extraction is the easiest and fastest way to use the representational power of pretrained deep networks. The network constructs a hierarchical representation of input image. Deeper layers contain higher-level features, constructed using the lower-level features of earlier layers. To obtain the feature representations of the training and test images, we use the last fully connected layer FC. Obtaining a lower-level representation of the images can be done by using an earlier layer in the network. In this chapter, various methodologies of copy-move forgery detection and image forensic detection were reviewed. Each study illustrated a specific approach to detect the image forgery. Some studies publicly shared their algorithms, while others did not. However, all of the mentioned researchers provided their own contribution to solve the problem statement of copy-move forgery detection within a limited capacity.

2.7 Conclusion

It is worth noting that the models mentioned above can be used in a patch size with good resolution. So, in order to achieve good matching with features from both patches, the extension on the feature length must remain lax (that is, in only a loosely extended state). Furthermore, we will apply sampling from cartesian and polar for ZM and PCT, respectively, whereas FMT will employ log-polar sampling. For the experiments, however, we will use polar sampling only to compute scaling

and rotation in order to obtain the optimized invariance angle as well as scalar values [54]. When we apply deep learning techniques, CNN architecture will be used for feature extraction by applying multiple layers with trainable filters to create feature maps which will be inserted later in row vectors. Technically, the deeper layers convey features of higher levels from the lower levels found within earlier layers. Then, feature extraction focuses on extracting useful information out of raw pixel values and use them in next processing steps.

2.8 References

[1] Rizvi, AB, " Digital Image Forgery Detection," 2015, Lincoln University, A New Zealand, 2015.

[2] Maryam Jaberi, George Bebis, Muhammad Hussain and · Ghulam Muhammad, "Accurate and robust localization of duplicated region in copy–move image forgery," *Machine Vision and Applications, Spriger*, vol. 25,pp. 451–475, 2014.

[3] Martin Harran, William Farrelly, Kevin Curran' " A method for verifying integrity and authenticating digital media," *Science Direct*, Applied Computing and Informatics. ,pp 145-158. 2018.

[4] Khanna N, Mikkilineni AK, and Delp EJ' " Forensic camera classification: verification of sensor pattern noise approach," 2009, Forensic Sci Commun Jan (11).

[5] Miroslav Goljan and Jessica Fridrich, " Identifying images corrected for lens distortion using sensor fingerprints," *In: Proc. SPIE, electronic imaging, media watermarking, security, and forensics XIV*,pp. 1-13, 2012.

[6] Shahzad Alam, Amir Jamil, Ankur Saldhi and Musheer Ahmad, " Digital Image Authentication and Encryption using Digital Signature ," *IEEE International Conference on Advances in Computer Engineering and Applications (ICACEA)* , pp. 332-336, 2015.

[7] Chiou-Ting Hsu and Ja-Ling Wu' " Hidden Digital Watermarks in Images," *IEEE Transaction on Image Processing*, vol 8. ,pp. 58-69, 1999.

[8] Ratnam Singh and Mandeep Kaur, " Copy Move Tampering Detection Techniques: A Review ," *International Journal of Applied Engineering Research*, vol. 11,pp. 3610-3615, 2016.

[9] J. Fridrich, D. Soukal, and J. Lukáš, "Detection of Copy-Move Forgery in Digital Images," *International Journal*, vol. 3,pp. 652-663, 2003.

[10] Lukas J., Fridrich J. and Goljan M, "Digital camera identification from sensor

pattern noise," *IEEE Transactions on Information Forensics and Security*, pp. 205-214, 2006.

[11] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva, " Efficient Dense-Field Copy–Move Forgery Detection," *IEEE Transactions on Information Forensics and Security*, Vol. 10, issue 11, pp. 2284-2297, 2018.

[12] Krizhevsky, A., Sutskever, I., Hinton, and G.E, " Imagenet classification with deep convolutional neural networks,"in *: Proceedings of the Advances in neural information processing systems*, pp. 1097-1105, 2012.

[13] Ren, S., He, K., Girshick, R., and Sun, J., " Faster r-cnn: Towards real-time object detection with region proposal networks," *In: Proceedings of the Advances in neural information processing systems*, pp. 91-99, 2015.

[14] He, K., Zhang, X., Ren, S., and Sun, J., " Identity mappings in deep residual networks," *In: arXiv preprint* , 1603.05027, 2015.

[15] Smeulders, A.W., Worring, M., Santini, S., Gupta, A., and Jain, R., " Content-based image retrieval at the end of the early years," *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, Vol. 12, pp. 1349-1380, 2000.

[16] Yang, J., Yu, K., Gong, Y., and Huang, T., " Linear spatial pyramid matching using sparse coding for image classification," *In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 20-25, 2009.

[17] Felzenszwalb, P.F., Girshick, R.B., McAllester, D., and Ramanan, D., " Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, Vol. 9, pp. 1627-1645, 2010.

[18] Jegou, H., Perronnin, F., Douze, M., Sanchez, J., Perez, P., and Schmid, C., " Aggregating local image descriptors into compact codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, Vol. 9, pp. 1704-1716, 2012.

[19] Zeiler, M.D., and Fergus, R., " Visualizing and understanding convolutional net-

works," *In: Proceedings of the European Conference on Computer Vision*, pp. 818-833. Springer, 2014.

[20] Simonyan, K., and Zisserman, A., " Very deep convolutional networks for large-scale image recognition," *In: arXiv preprint* , arXiv:1409.155 , 2015.

[21] He, K., Zhang, X., Ren, S., and Sun, J., " Deep residual learning for image recognition," *In: arXiv preprint*, arXiv:1512.03385, 2015.

[22] Xie, S., Girshick, R., Dollar, P., Tu, Z., and He, K., " Aggregated residual transformations for deep neural networks," *In: arXiv preprint*, arXiv:1611.05431 , 2016.

[23] S. Das, " CNNs Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more," Medium, 2017.

[24] Jing, W., Hongbin, Z. "Exposing digital forgeries by detecting traces of image splicing," in 8th International Conference on Signal Processing, vol. 2, IEEE, 2006.

[25] Christlein, V., Riess, C., Jordan, J., Riess, C., Angelopoulou, E. "An evaluation of popular copy-move forgery detection approaches," IEEE Transactions on Information Forensics and Security 7(6), pp. 1841-1854, 2012.

[26] Ryu, S.J., Kirchner, M., Lee, M.J., Lee, H.K, "Rotation invariant localization of duplicated image regions based on Zernike moments," in. IEEE Transactions on Information Forensics and Security, pp. 1355-1370, 2013.

[27] Haodong Li, Weiqi Luo, Xiaoqing Qiu, and Jiwu Huang," IEEE Transactions on Information Forensics and Security, vol. 12, no. 5, pp. 1240– 1252, 2017.

[28] Pawel Korus and Jiwu Huang, "Multi-scale analysis strategies in prnu-based tampering localization," IEEE Translated Transactions on Information Forensics and Security, vol. 12, no. 4, pp. 809–824, 2017.

[29] Lee, H., Ekanadham, C., Ng, A.Y, "Sparse deep belief net model for visual area," In: Advances in Neural Information Processing Systems, 20 (NIPS), 2008.

[30] Larochelle, H., et al, "Exploring strategies for training deep neural networks," J.

Mach. Learn. Res. 10(10), pp. 1–40, 2009.

[31] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P, "Gradient-based learning applied to document recognition," Oroc. IEEE, pp. 2278-2324, 1998.

[32] Giacinto, G., Roli, F., "Design of effective neural network ensembles for image classification purposes," Image Vis. Computer, pp. 699-707, 2001.

[33] Fukushima, K., Miyake, S., Ito, T., "Neocognitron: a neural network model for a mechanism of visual pattern recognition," IEEE Trans. Syst. Man Cybern, pp. 826–834, 1983.

[34] Andrea V., K. L., Ankush Gupta, "Convolutional Neural Networks for Matlab," MatConvNet, pp. 1-59, 2015.

[35] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9, 2015.

[36] B. Bayar, M. C. Stamm, "Design principles of convolutional neural networks for multi-media forensics," Society for Imaging Science and Technology, pp. 77–86, 2017.

[37] L. Bondi, L. Baroffio, D. Guera, P. Bestagini, E. J. Delp, and S. Tubaro, "First Steps Toward Camera Model Identification with Convolutional Neural Networks," IEEE Signal Processing Letters (SPL), 24: pp. 259–263, 2017.

[38] Gregory Koch, " Siamese Neural Networks for One-Shot Image Recognition," Computer Science Dep. University of Toronto ,pp. 1-30, 2015.

[39] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde- Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in neural information processing systems, 2014, pp. 2672–2680.

[40] Wu, Y., Abd-Almageed,W., Natarajan, P. "Deep matching and validation network: An end-to-end solution to constrained image splicing localization and detection." In: Proceedings of the 2017 ACM on Multimedia Conference. pp. 1480–1502.

MM '17, 2017.

[41] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in IEEE ICCV, 2017.

[42] T. A. B. X. Han, " MatchNet: Unifying Feature and Metric Learning for Patch-Based Matching.," Proceedings of Computer Vision and Pattern Recognition, 2015.

[43] Im, D.J., Kim, C.D., Jiang, H., Memisevic, R.: Generative adversarial metric, 2016.

[44] Vanita Agarwal and Vanita Mane, " Reflective SIFT for Improveing the Detection of Copy-Move Image Forgery,"*Second International Conference on Research in Computational Intelligence and Communication Networks* , (ICRCICN) ,pp. 84-88, 2016.

[45] Ira Tuba, Eva Tuba, and Marko Beko, " Digital Image Forgery Detection Based on Shadow Texture Feature,"*Telecommunications Forum 2016(16th)*, (TELFOR),pp. 22-23, 2016.

[46] Zhuo Yang, and Ser. Kamata, " Fast Polar Cosine Transform for Image Description," *MVA2011 IAPR Conference on Machine Vision Applications, Nara, JAPAN*,pp. 320-323, 2011.

[47] Teague' and MichaelReed, " Image analysis via the general theory of moments," *Journal of the Optical Society of America*, vol. 70,pp. 920-930, 1980.

[48] Avidan, Igor Olonetsky and Shai, " TreeCANN - k-d tree Coherence Approximate Nearest Neighbor algorithm," *Proc. 12th Eur. Conf. Comput. Vis*, pp. 602-615, 2012.

[49] L. D'Amiano, D. Cozzolino, G. Poggi and L. Verdoliva, " Video forgery detection and localization based on 3d patchmatch," Multimedia and Expo Workshops ICMEW, *IEEE International Conference on*, pp. 1-6, 2015.

[50] Sheehan, Brady, " Analysis of Copy-Move Forgery Detection," 2015, Duquesne University, Pittsburgh.

[51] Giovanni Chierchia, Giovanni Poggi,Carlo Sansone and Luisa Verdoliva, " A Bayesian-MRF Approach for PRNU-Based Image Forgery Detection," *IEEE Transactions on Information Forensics and Security*, Vol. 9, pp. 554-567, 2014.

[52] Yuan-Neng Hsu, H. H. Arsenault, and G. April, " Rotation-invariant digital pattern recognition using circular harmonic expansion," *OSA Publishing*, Vol. 21, pp. 4012-4015, 1982.

[53] Seung-Jin Ryu, Matthias Kirchner, Min-Jeong Lee, and Heung-Kyu Lee, " RotationInvariantLocalizationofDuplicatedImage Regions Based on Zernike Moments," *IEEE Transactions on Information Forensics and Security*, Vol. 8, pp. 1355-1370, 2013.

[54] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva, " Efficient Dense-Field Copy–Move Forgery Detection," *IEEE Transactions on Information Forensics and Security*, Vol. 10, issue 11, pp. 2284-2297, 2018.

[55] S. Sadeghi, S. Dadkhah, Hamid A. Jalab, G. Mazzola and Diaa Uliyan , " State of the art in passive digital image forgery detection: copy-move image forgery," *springer, Pattern Analysis and Applications*, Vol. 10, issue 21, pp. 291–306, 2018.

Chapter 3

A Robust Method for Rotated and Scaled Copy-Move Forgery Detection Using Enhanced PatchMatch

Preface.

A version of this chapter has been published in the Journal of computer science issues 2017; 14: ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784: 1-7. I am the primary author. Along with Co-authors, Tariq Iqbal and M. Shehata, I developed the conceptual algorithm. I have carried out most of the data collection and analysis. I have prepared the first draft of the manuscript and subsequently revised the manuscript, based on the feedback from Co-authors and also peer review process. As Co-authors, Tariq Iqbal and M. Shehata assisted in developing the concept and testing the algorithm, reviewed and corrected the model and results. They also contributed in reviewing and revising the manuscript. Contribution was made through the team sup-

port to developing and testing the algorithm as well as and reviewing the manuscript.

Abstract - Image forgery detection approaches are varied and serve same objectives. However, the difference in image properties causes some limitations of most of these approaches. Integrate multiple forensic approaches to increase the efficiency of detecting and localize the forgery was proposed based on the same image input source. In this paper, we propose a new detector algorithm based on different image source format. We propose approach to detect a copy-move forgery based on PatchMatch enhanced by the dense field technique. The F-measure score used same evaluation function to make the system more robust. The output result shows high efficiency of detecting and localizing the forgery in different image formats, for passive forgery detection.

Keywords: Copy-move detection; forgery localization; image forgery; score evaluation.

3.1 Introduction

The software and hardware technologies reduce the gap between the professional people and the amateurs in different fields. Digital image processing, computer graphics, and computer vision have some advantages and disadvantages to the use of technology. Forgery is one of the challenging issues of digital image processing in recent decades. As a result of using new algorithms and investigation techniques, it becomes possible to detect the forgery [1]. Image forgery detection approaches are varied and serve the same objectives. However, the difference in image properties causes some

limitations of most of these approaches. Integrate multiple forensic approaches to increase the efficiency of detecting and localize the forgery was proposed based on the same image input source. In this work, we propose a new detector algorithm based on rotation and scaling invariance methods. We propose a robust approach to detect a rotated and scaled copy-move forgery based on PatchMatch enhanced by the dense field technique. The F-measure score used the same evaluation function to make the system more robust. The output result shows the high efficiency of detecting 91% and localizing the forgery in different image formats.

3.2 PatchMatch Copy-move Forgery Detection Approach

A PatchMatching algorithm is faster than most other algorithms and uses a matching approach that applies dense approximation field matching. The primary reason for choosing this approach over others is its fast propagation in offset fields. Iterations are performed by performing a randomized search or by doing full-image scanning (propagation). Generally, in a scan, we can first choose a specific vector $f(s)$, that utilizes an s pixel for its patch center. Because the features essentially characterize the patch, the distance between and among the features need to be carefully and accurately measured.

.

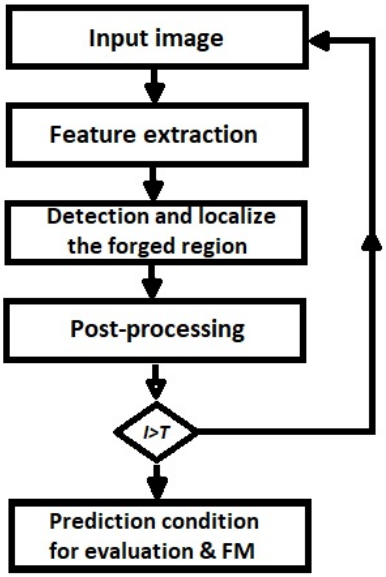

Figure 3.1: A PatchMatching-based copy-move forgery detection algorithm

3.3 Proposed Method

Digital image copy-move forgery detection using PatchMatch has many limitations to deal with most of geometric transformation which normally applies to copy-move forgery, such as rotation, scaling and even shearing of the copied region. PatchMatch was enhanced by [2] to deal with rotation, scaling and could be extended to more geometric transformations. The algorithm complexity increased by considering the scale factor and the rotation angle as new dimensions in the space. Both rotation and scaling's offset fields are changed in a linear manner. This theory, make the propagation stage done easier and quicker as whenever the offset field matching in the same neighborhood. However, the used enhanced PatchMatch algorithm works very similarly to the classic version of this algorithm [2]. In fact, the proposed approach achieved more robustness by improving a random search to overcome the local minima trap that results of a large matching scale with high dimensional optimization space. Reducing the number of iterations by initializing offset point increased the speed of the algorithm. Figure 3.2 shows the matching of offset points over varies of geometric

transformations.

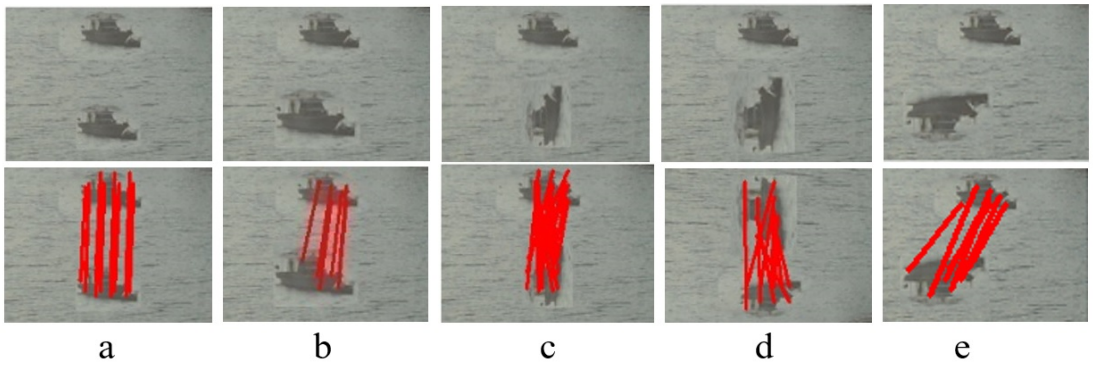

Figure 3.2: Matching of the offsets in copy-move forged images in: a. street forward copy-move, b. scaled copy-move, c. rotated copy-move, d. rotated and scaled copy-move, e. rotated and scaled in mirrored copy-move)

.

3.3.1 Post-Processing using Denes Liner Fitting

Comparison of images, along with matching and stitching, is more or less based on feature matching. To create an offset field, the PatchMatch algorithm employs matching and a feature search using offset points and generates the offset field. In this system, a linear offset will formulate an accurate offset field on the top of the copy-move region. Referred to as "propagation", this step might require several iterations. Dense-field matching strategies enhance the efficacy of the strategy and thus have already been the choice of many scholars, as shown in [3-6]. Yet, despite the method's popularity, this type of image can suffer from geometric deformation, compression, the noise effect, and illumination fluctuations, and geometric deformation. When this occurs, the offset field is significantly less successful at feature-matching. In the post-processing stage, the objective is to remove or at least mitigate all negative impacts on the image.

To achieve this aim, it is important to regularize offset fields and thereby heighten the opportunity to enhance the detection of copy-move and decrease false alarms. To be viable, offset fields must be able to fit all neighborhood pixels of s through a linear model, after which a transformation sets parameters to obtain the sum of square error (SSE).

$$\delta^{'}(s_i) = AS_i \tag{3.1}$$

$$\epsilon^2(S) = \sum_{i=1}^{N} \|\delta(s_i) - \delta^{'}(s_i)\|^2 \tag{3.2}$$

.

In the post-processing strategy, a number of steps need to be adhered to for the tests to be valid: 1) filtering median material using a circular window with a radius of ρ_M; 2) computing fitting errors, $\epsilon^2(S)$, w.r.t. using a least-squares linear model over a circular neighborhood of radius ρ_N; 3) bringing $\epsilon^2(S)$ to level T_ϵ^2 ; 4) deleting regional couples that are actually closer positioned than T_{D2} pixels; 5) deleting of all regions not larger than T_S pixels; 6) mirroring the regions; and 7) morphological dilation of the elements using a circular structuring element featuring a radius of $\rho_D = \rho_M + \rho_N$. If we choose to use these clearly defined stages, our first step is to get rid of all outliers from the image using a median filter. In fact, not until all of the outliers have been deleted or demoted will the minimum mean square fitting be used. Pictures and other images that exhibit repeating patterns, including monochrome, can be extremely problematic because their identical or near-identical details can lead to mismatching entire areas. To overcome this problem, we use the thresholds T_ϵ^2 , T_{D2}, and T_S, whose usage is indicated in steps 3, 4, 5. So, if a copy-move pixel is suspected, s, for a particular area, the mirrored pixel "twin" in $s + \delta(s)$ is designated as a copy-move pixel. The final stage will view morphological effects as an outcome

of immediately preceding steps.

3.3.2 F-measure Procedure

The F-measure standing (i.e., score) within the IEEE designation is determined by a procedure that determines the true condition for negative and positive conditions. This must have in it both pixel- and image-level images. Table 3.1 categorizes all conditions accordingly to the scoring outcome.

Table 3.1: The predictive conditions

	Predictive conditions		
	Predictive Positive		Predictive Negative
TP	$TPR = \frac{TP}{\sum condition - positive}$	FN	$FNR = \frac{FN}{\sum condition - positive}$
FP	$FPR = \frac{FP}{\sum condition - negative}$	TN	$TNR = \frac{TN}{\sum condition - negative}$

The accuracy of any approach depends on all area under the PDF curve. Based on that, we can write the equation as follows

$$Acc = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.3}$$

.

In brief, true positive shows the highest-output standings regarding accuracy in detecting forged images, and true negative shows zero-output tallies featuring zero scores. This indicates that the approach has detected zero evidence of forgery. Conversely, false negative and false positive indicate, respectively, a non-detected forged image and an image that was assumed false, but the assumption was inaccurate. For CMFD, the output can indicate a pure image mask or a forgery. On the other hand, the ground truth mask presents as a binary mask (0, 1) which can be created by

hand to signify the region copies as well as removing the high-value image elsewhere within the same shot (groundroot==max). In this instance, the remaining mask is considered a low value (groundroot==0). Once the condition values are obtained, we can verify the process, assuming that the CMFD output/ground root constitutes genuine inputs. In the initial test, we can make both inputs the same to obtain a valid F-measure score. Next, we can apply various inputs to achieve a range of F-measures depending on the predicted condition values. Figures 3.3 to 3.4 indicate F-measure results from the inputs. In this work, the "ideal" value for F-measure will be the outcome of an ideal matching of the ground root mask (GT) and the output mask of forgery detection function. In so doing, outliers of CMFD will likely lead to low F-measure and thus limit the validity of the system.

Figure 3.3: If the CMFD mask and the GT mask are the same, the F-measure is considered ideal

.

Figure 3.4: If the CMFD and GT masks turn variant, the F-measure is invalidated)

.

3.4 Experiment Result

The proposed algorithm was able to resolve all the issues presented above and successfully apply the most apt forgery detection examples for detecting copy-move and localizing it. However, there are situations when the process will be less efficient, especially if the image contains too many vivid colors, no colors, or is in black and white. Overall, the detector process requires the use of a variety of images and datasets, including the Loughborough University dataset [1] and the GRIP database [2].

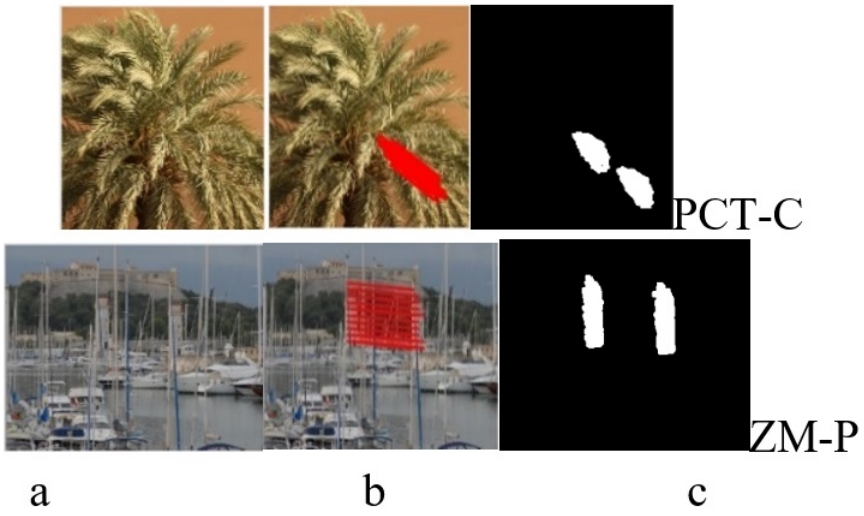

Figure 3.5: Searching for forged RGB images: (a) forged image, (b) offset points, (c) localization copy-move forgery mask)

.

.

.

[1] http://www.grip.unina.it.

[2] http://homepages.lboro.ac.uk/ cogs/datasets/ucid/ucid.html.

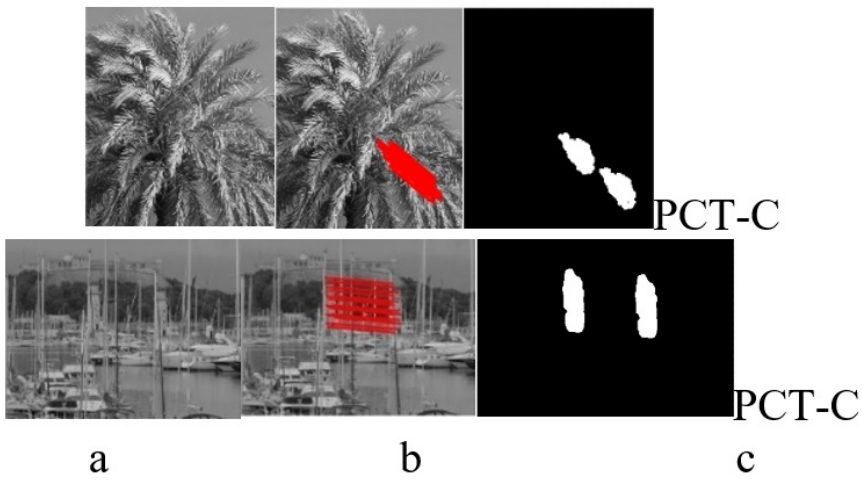

Figure 3.6: Searching for forgery in a grey image: (a) forged image, (b) offset points, (c) localization copy-move forgery mask)

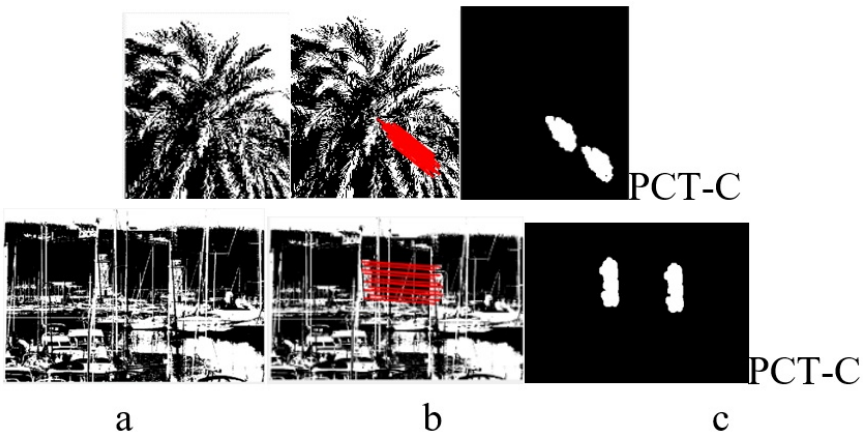

Figure 3.7: Searching for forgery in black and white image: (a) forged image, (b) offset points, (c) localization copy-move forgery mask)

In searching the three instances given in Figures. 3.5 to 3.7, we see several different offset points and also notice the reduction in efficiency when there are fewer colors. Moreover, when presented from exceedingly "flat" viewpoints, changing image formatting to BW from RGB could lead to even further reductions in features, which means that the forgery, if present, might not be viewable with the tools provided.

Applying the proposed algorithm on digital images which were targeted by scaling and rotation manipulations, shows that the robustness of this scheme is excellence in the matter of detecting and localizing the forgery. Figures 3.8 to 3.11 show some result of such experiments. (see Appendix A for more figures).

Figure 3.8: Detecting forgery in rotation by 90 degree)

.

Figure 3.9: Detecting forgery in rotation by 180 degree)

.

.

.

In order to objectively evaluate the usefulness and applicability of the methods, we decided to apply the identical function as outlined by researchers in [6][7]. In

Figure 3.10: Detecting forgery in rescaling by 80 percent)

Figure 3.11: Detecting forgery in rescaling by 120 percent)

examining Table 3.2, however, it becomes clear that the conditions are different. The proposed method was used to detect active and passive copy-move forgery in different images format, while the PRUN method, for example, was only used for active copy-move forgery detection using a specific dataset that dragooned the proposed method accuracy to be lower than that method. However, in the end, they gave very close FM value. The algorithm was examined in many different images. Indeed, the proposed algorithm robustness was judged against passive detector Discrete Cosine Transformation DCT (Sheehan 2015), and the active detectors in literature like photo response non-uniformity PRNU (Giovanni Chierchia 2014) when the original image was presented as in figure 3.12.(See Appendix). Table 3.3 shows the time complexity of the proposed method and the DCT method form literature. Here the comparison shows the total time for full image inspection. The DCT algorithm using a block scanning level, while the proposed method used pixel and feature propagation level. That, in fact, makes the DCT algorithm take less time in most inspection cases. The table shows one example. The difference of the FM in these experiments is negligible. Tables 3.4 and 3.5 show some results of these experiments.

Table 3.2: FM Parameters of Image as Given by the Proposed Method vs. Some Literature

	Existing methods			Proposed method	
Par.	SIFT	DyWT+SIFT	PRNU	PatchMatch	E-PatchMatch
TPR	0.953	0.888	0.960	0.539	0.952
TNR	0.791	0.818	0.978	0.993	0.994
FPR	0.046	0.111	0.039	0.460	0.046
FNR	0.029	0.091	0.003	0.105	0.004
Acc	85.5%	85%	97.7%	90.88%	90.5%
FM	0.837	0.842	0.88	0.6885	0.928

Table 3.3: Time complexity of the proposed algorithm vs literature

Used Algorithm	Proposed method PatchMatch	Existing method DCT
Time	302.800742 seconds	132.146030 seconds

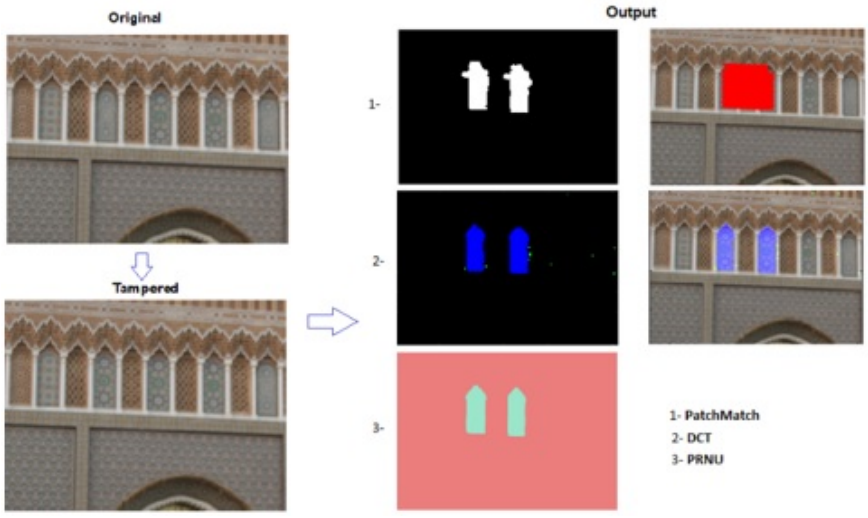

Figure 3.12: Output of the proposed detector vs active and passive detectors from literature)

Overall, the proposed approach achieved more robustness by 1) improving a random search to overcome the local minima. 2) Reducing the number of the iterations by initializing offset point 3) Increased the speed of the algorithm.

Table 3.4: This table shows the proposed evaluation values for detecting CMF in four different RGB images.

Input-img-No.	FM	TPR	TNR	FNR	FPR	PPV	NPV	TFE	TPM	TPP
1.	0.999	0.9958	0.9995	0.0042	0.0005	0.9862	0.999	1.292	12.235	1.465
2.	0.9992	0.9987	1.0000	0.0013	0.00001	0.9997	1.0000	1.945	10.179	1.687
3.	0.9727	0.9972	0.9977	0.0028	0.0023	0.9493	0.999	1.912	10.871	1.703
4.	0.5633	0.7210	0.9683	0.2790	0.0317	0.4622	0.9892	1.892	11.131	1.753

Table 3.5: This table shows the evaluation values for detecting CMFD to same image in assorted color format.

Feature-Type	FM	TPR	TNR	FNR	FPR	PPV	NPV	TFE	TPM	TPP
PCT-BW	0.9896	0.9905	0.9996	0.0095	0.0004	0.9888	0.9997	1.230	8.765	1.579
PCT-RGB	0.999	0.9958	0.9995	0.0042	0.0005	0.9862	0.999	1.292	12.235	1.465
ZM-Gray	0.9802	0.9733	0.9996	0.0267	0.00049	0.9873	0.9990	2.060	11.657	1.790

.

3.5 Conclusion

To conclude, copy-move forgery detection (CMFD) had been widely adopted for use by people of all skill levels, due mainly to its user-friendly and ease-of-use approach. However, despite the relative simplicity of the strategy, there are still some challenges that go along with it that makes the outcome sometimes invalid or at least questionable. On the whole, there is the main issue affecting most CMFD algorithms. If a

copy-move is performed by applying something in the image background to obscure evidence of forgery, but this can be overcome by employing classic PatchMatching on the forged images' offset points. However, applying a geometric transformation on a forged digital image, like scaling or rotation, will be more challenging. In this situation, the authentic image is needed to proceed with forgery detection, so a different method should be adopted. Our experiments indicate the presence of variance within the evaluations, which occurs also in identical images where there are alterations to the resolution or color, giving unequal F-scores. Despite these slight problems, the F-score generally exhibits optimal efficiency in the enhanced approach. In future studies, we would use the identical idea to test for forgeries in videos and include studies from the literature to evaluate F-score results.

3.6 References

[1] Anil Dada Warbhe, Rajiv V. Dharaskar, and Vilas M. Thakare, “Block Based Image Forgery Detection Techniques,” *International journal of engineering science and research technology*, pp. 289-297, 1997.

[2] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, “The generalized PatchMatch correspondence algorithm,” in Proc. 11th Eur. Conf. Comput. Vis., 2010, pp. 29–43.

[3] S. K. a. S. Avidan, "CoherencySensitiveHashing," in Proc. Int. Conf. Comput. Vis., p. 1607–1614, 2011.

[4] I. O. a. S. Avidan, "TreeCANN - k-d tree Coherence Approximate Nearest Neighbor algorithm," Proc. 12th Eur. Conf. Comput. Vis, p. 602–615, 2012.

[5] D. C. G. P. a. L. V. L. D’Amiano, "Video forgery detection and localization based on 3d patchmatch," Multimedia and Expo Workshops (ICMEW), 2015 IEEE International Conference on, pp. 1-6, 2015.

[6] G. P. a. L. V. Davide Cozzolino, "Efficient Dense-Field Copy–Move Forgery Detection," IEEE Transactions on Information Forensics and Security, vol. 10, no. 11, pp. 2284-2297, 2015.

[7] J. Besag, "On the statistical analysis of dirty pictures," Journal of the Royal Statistical Society, Series B 48, p. 259–302, 1986.

Chapter 4

Convolutional Neural Network for Copy-move Forgery Detection

Preface.

A version of this chapter has been published in the Journal of Symmetry,MDPI 2019; 11: Isue 10, 10.3390/sym 11101280, 1-17. I am the primary author. Along with Co-authors, Tariq Iqbal and M. Shehata, I conceptualized the idea. I have prepared the first draft of the manuscript and subsequently revised the manuscript, based on the feedback from Co-authors and also peer review process. As Co-authors, Tariq Iqbal and M. Shehata assisted in developing the concept and testing the algorithm, reviewed and corrected the model and results. Also, the Co-author T. Iqbal critically reviewed the content and revising the manuscript. Contribution was made through the team support to developing and testing the algorithm as well as and reviewing the manuscript.

Abstract: Digital image forgery is a growing problem due to the increase in readily-available technology that makes the process relatively easy. In response, several approaches have been developed for detecting digital forgeries. This paper proposes a

novel scheme based on neural networks and deep learning, focusing on the convolutional neural network (CNN) architecture approach to enhance a copy-move forgery detection. The proposed approach employs a CNN architecture that incorporates pre-processing layers to gives satisfactory results. In addition, the possibility of using this model for various copy-move forgery techniques is explained. The experiments show the overall validation accuracy is 90%, with a set iteration limit.

Keywords: forgery detection; neural networks; image processing.

4.1 Introduction

Digital editing is becoming less and less complicated with time thanks to the increased availability of a wide array of digital image editing tools. Image forgery, which is defined as "the process of cropping and pasting regions on the same or separate sources" [1], is one of the most popular forms of digital editing. Copy-move forgery detection technology can be applied as a means to measure an image's authenticity. This is done through the detection of "clues" that are typically found in copy-move forged images. In the field of digital image forensics, copy-move forgery detection generally falls into two categories: keypoint-based and block-based [2]. This chapter will focus on the latter category. Block-based copy-move forgery detection approaches employ image patches that overlap. From these, "raw" pixels are removed for forgery testing against similar patches [3].Of the many strategies currently being employed in image forgery detection, several use statistical characteristics across a variety of domains [4]. Regardless of the forgery category, the forgery detection application will deal with active image copy-move forgery and/or passive copy-move forgery. In the former

type, the original image includes embedded valuable data which makes the detection process easier whereas in the latter the original is imaging which makes the detection more challenging and difficult. Image forgery localization is even more difficult to carry out [5]. While forgery detection only seeks to know if an image is in whole or in part fake or original, image forgery localization tries to find the exact forged portions [5][6]. Furthermore, in image forgery localization, the focus is on building a model rather than looking at only certain features or domains. The model will be used to automatically detect specific elements based on a form of advanced deep neural network. Examples of these types of networks include Deep Belief Network [7], Deep Auto Encoder [8], and Convolutional Neural Network (CNN) [62]. Of these three neural networks, CNNs are most commonly used in vision applications. These approaches employ local neighborhood pooling operations and trainable filters when testing raw input images, thereby creating hierarchies (from concrete to abstract) of the features under examination. Because the image analysis and computer vision in the CNN strategy are so highly advanced, CNN generally provides excellent performance [10-12] in image forgery detection through the composition of simplistic non-linear and linear filtering operations (e.g., rectification and convolution) [13]. This present paper proposes a novel approach for image forgery detection and localization which is based on Scale Variant Convolutional Neural Networks (SVCNNs). An outline of the proposed method is presented in Figure 4.3. For this approach, sliding windows that incorporate a variety of scales are included in customized CNNs with the aim of creating possibility maps that indicates image tampering. Our main focus is both copy-move forgery detection and localization through the application of elements removed via the use of CNNs. The rest of the paper is organized as follows. In section 2, we introduce an overview of the literature that has contributed to the advancement of CNNs in copy-move forgery detection and feature extraction proce-

dures. In section 3, we introduce the proposed model and the training processes. In section 4, the experiment's environment and results are discussed. Finally, in section 5, we present the study's conclusions.

4.2 Related Work

This section provides an overview of related works in copy-move forgery detection using neural network CNN's and related concepts. CNN for forgery detection based on DCT: Numerous researchers have approached the problem using CNN's for forgery detection. As discussed in [14], CNNs can be used in steganalysis for gray-scale images, where the CNNs first layer features a single high pass filter to filter out the image content. In [2], an image model is developed for detecting image-splicing forgery. In this approach, the researchers used discrete cosine transformation (DCT), to remove relevant features out of the DCT domain [2].

The DCT domain feeds the input of the CNN by transferring the row of quantized DCT coefficients from the JPEG file to data classification. The processing of the data in the classification stage will generate a histogram for each patch and concatenate all the histograms to feed the CNN [36]. In [15-17], deep learning methods applied to computer vision problems resulted in a local convolution feature data-driven CNN, while in other research, copy-move forgery detection algorithms were mostly based on computer vision tasks like image retrieval [18], classification [19], and object detection [20]. Along with CNN, GPU technologies have helped to fuel the latest improvements in computer vision tasks [15]. Unlike traditional strategies for image classification, which mostly use local descriptors [21], the latest CNN-based image classification techniques use end-to-end structure. Because deep networks typically incorporate classifiers and features that are high, mid, or low level [22] using end-to-end multilay-

ers, the various feature levels are enriched according to the number of hidden layers. The most recent convolutional neural networks (e. g., VGG [23], AlexNet [15], ResNet [14][17] and ResNeXt [24]) significantly enhance performance in object detection and image classification tasks [16]. Table 4.1 provides a brief summary of some common CNNs [26].

Table 4.1: The common CNNs characteristics

No.	CNN	Layer No.	Inventer(s)	Year	Place	Parameters No.	Error rate
[25]	LeNet	8	Yann LeCun	1988	First	60 T	N/A
[66]	AlexNet	7	Alex K. Hinton, Ilya S.	2012	First	60 M	15.5%
[76]	ZFNet	7	Matthew Z, Rob Fergus	2013	First	N/A	14.8%
[25]	Google Net	9	Google	2014	First	4 M	6.67%
[70]	VGG Net	16	Simonyan, Zisserman	2014	Second	140 M	3.6%
[72]	ResNet	152	Kaiming He	2015	First	N/A	3.75%

In the CNNs mentioned above, the intermediate layers serve as global features of image-level descriptors. This type of feature can reinforce inter-class differences but does not make any intra-class distinctions. The strategies for deep learning applied to computer vision tasks are also not suitable for direct use in copy-move forgery detection. As discussed previously, this kind of detection looks for the same types of regions that have been resized, rotated or deformed in some way. The expressive feature representations output derived from image-level CNNs [27] points to the possibility of using appropriate patch-level descriptors in order to replace handcrafted patch-level descriptors with data-driven ones [27]. The recent literature presents a number of deep local descriptors that offer impressive patch classification and matching abilities [28]. Because CNNs have been proven proficient in natural image distribution, they will likely also be useful in image copy-move forgery detection, given that the aim in that task is to find the so-called natural or pristine image among any unnatural or forged ones. The main key used to classify images in order to detect copy-move

forgery and localize it is image features. Therefore, extracting image features is an essential part of the CNNs' work in copy-move forgery detection. We will highlight the differences between the classic way and the CNN automatic way of feature extraction and show how the CNN strategy effectively eliminates the need for the first method.

4.2.1 Feature Extraction

The literature includes several different feature extraction approaches. Although the published works discuss a wide range of different strategies for detecting copy-move forgery, the present study will focus on three specific classifications of features, which are the polar cosine transforms (PCT), the Zernike moments (ZM), and the Fourier-Mellin transform (FMT). These three techniques were explained in previous chapter (3).

4.2.2 Using CNNs for feature extraction

In neural networks, the process of feature extraction removes elements of learned images out of a pre-trained CNN (see Figures 4.1 and 4.2). These images can then be utilized for training image classifiers. In general, feature extraction presents as the simplest approach when applying pretend deep networks of representational power as there is a clearly delineated hierarchy of the input images which is easy to understand. In short, the deeper layers convey features of higher levels and are built by incorporating features from the lower levels found within earlier layers. Test and training images for feature representations can be sourced from previous fully-connected (FC) layers, while image representations from lower levels require an earlier network layer.

.

.

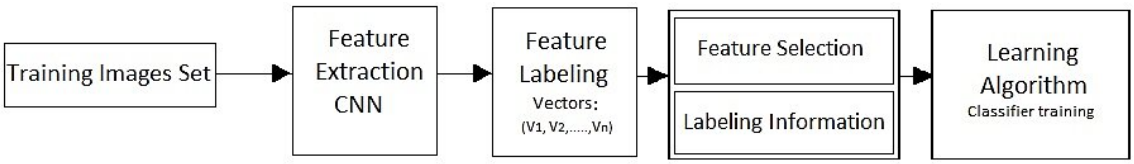

Figure 4.1: Feature selection for classification pipeline (The proposed model). Training Stage

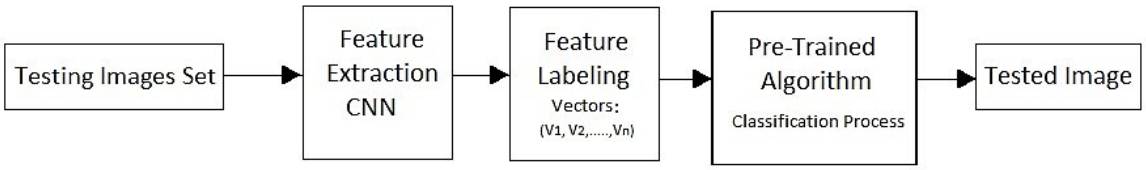

Figure 4.2: Feature selection for classification pipeline (The proposed model). Testing Stage)

4.2.3 Classifying feature selections

The last step for computer vision applications is to use feature selection in object identification to classify certain features according to specific characteristics. This stage is typically carried out using later layers of deep learning neural networks via a voting technique. Take, for example, the fully-connected layer known as learning. Because of a large amount of data, it can be challenging for the system to learn good classifiers prior to extracting undesirable features from the program. However, removing features which are irrelevant or repetitious creates a more generally applicable classifier and also serves to decrease learning algorithm run times, thus enabling a deeper understanding of the real-world problem to which the classifier is being applied.

A considerable number of worthy research has already been conducted on existing techniques for detecting and localizing copy-move forgeries. The research has investigated whether these implemented methods are sufficiently robust and whether properly modeling the structural changes that have occurred in images due to copy-move forgeries can reliably classify a digital image as a pristine or manipulated image.

Furthermore, many techniques of copy-move forgeries have been presented in the literature. Some good examples about detections techniques and their limitations [29-31]. Some recent studies [29][32] on copy-move forgery detection highlighted CNN's which learn and minimize a loss function (an objective that scores the quality results) in an automatic process. However many authors are still attempting manual efforts on designing effective loss function by telling CNN what we wish to minimize [31][33].

4.3 The Proposed CNN Model

CNN's are nonlinear interconnecting neurons based on the construct of the human visual system. Applying CNN's for forensic purposes is a somewhat new approach, but their ability to segment images and identify objects is thus far unsurpassed [34]; In one study where CNNs were used to extract input images' features in order to classify them the method outperformed all previous state-of-the-art approaches. The proposed CNN will be used as a feature extractor for image input patches in the training stage and, later on, for the testing stage as well (see Figures 4.1 and 4.2). CNN's can be deconstructed into building blocks knowns as layers. Layer L_i will accept relevant input $H_i \times W_i \times P_i$ for feature maps or vectors sized as P_i . This layer then gives the output $H_{i+1} \times W_{i+1} \times P_{i+1}$ for feature maps or vectors sized as P_{i+1}. In the present study, we use six different kinds of layers: convolutional, pooling, ReLU, softmax, fully-connected, and batch normalization. A brief description of each type of these layers is given below.

1. In a convolutional layer, the convolutions are performed using stride S_h and S_w for the first two axes of the input feature maps, along with P_{i+1} filters $K_h \times K_w \times P_i$ [35].

$$H_{i+1} = \left[\frac{H_i - K_h + 1}{S_h}\right] \tag{4.1}$$

$$W_{i+1} = \left[\frac{W_i - K_w + 1}{S_w}\right] \tag{4.2}$$

$$P_{i+1} \tag{4.3}$$

2. In a pooling layer, which occurs following convolutions, the layer chooses pixel valuations of specific characteristics (e.g., average pooling or maximum pooling) within a given region. If a max-pooling layer is chosen, it then carries out maximum element extraction, i.e., stride S_h and S_w for the initial two axes in a neighborhood $K_h \times K_w$ for every two-dimensional piece of input the feature map. The input block's maximum value is, therefore, returned [35]. This approach is commonly applied in deep learning networks. In our proposed strategy, the max-pooling layer will decrease the input image patch resolution as well as enhance network robustness in the face of possible valuation changes in the motion residuals of the frame's absolute difference image [36].

$$H_{i+1} = \left[\frac{H_i - K_h + 1}{S_h}\right] \tag{4.4}$$

$$W_{i+1} = \left[\frac{W_i - K_w + 1}{S_w}\right] \tag{4.5}$$

$$P_{i+1} = P_i \tag{4.6}$$

Input image patches for CNN models use two-dimensional array image blocks measuring $3 \times (64 \times 64)$, with 3 indicating the channel number in the RGB-scale. Thus, if we use 3×3 as a window size and 3 as the stride size, the image patch resolution decreases by half to 32×32 from its original 64×64 following the initial max-pooling layer [35].

3. ReLU layer Performs element-wise nonlinear activation. Given a single neuron x, it is transformed into a single neuron y with:

$$y = max(0, x) \tag{4.7}$$

4. Softmax layer Turns an input feature vector to a vector with the same number of elements summing to 1. Given an input vector x with P_i neurons $x_j i \in [1, P_i]$, each input neuron produces a corresponding output neuron.

$$y_i = \frac{e^{xj}}{\sum_{k=1}^{k=p_i} e^{xk}} \tag{4.8}$$

5. In a fully-connected (FC) layer, dot multiplication is carried out between flattened feature maps (i.e., the input feature vector) and the weight matrix using P_{i+1} rows, along with columns of $P_i or(H_i.W_i.P_i)$. Meanwhile, the output feature vector presents $P_(i+1)$ elements. Trained CNNs can also remove meaningful information in images that have not been used to train the network. This particular characteristic enables forgery exposure of previously unidentified images as well [35].

6. In a batch normalization layer, every input channel is normalized in ultra-small (or mini) batches. The batch normalization layer initially normalizes every individual channel's activations by subtracting the mini-batch mean and then dividing the result by the standard deviation of the mini-batch. Next, the input is shifted by the layer using the learnable offset β, after which it scales the input using the learnable scale factor γ . Batch normalization layers can also be used between convolutional and nonlinearities (e.g., ReLU layers) to increase CNN training and lessen any sensitivities that might arise during the initialization of the networks. Batch normalization can normalize inputs x_i through formulating the mean μ_B and variance σ_{B^2} for a mini batch and input channel, after which it formulates the normalized activations:

$$\widehat{x_i} = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{4.9}$$

As can be seen in the expression above, ϵ or Epsilon is used to enhance numerical stability if the variance of the mini-batch variance presents as being too small. Furthermore, in cases where the zero mean input and unit variance are not suited to the subsequent batch normalization layer, it is then scaled and shifts its activations as follows:

$$y_i = \widehat{\gamma x_i} + \beta \tag{4.10}$$

Interestingly, the offset β and scale factor γ properties appear as learnable properties that can be updated throughout the network training process. At the end of the network training, the batch normalization layer then formulates both the mean and the variance across the entire training set, after which it retains them as properties called TrainedMean or TrainedVariance [35]. Then, if the trained network is applied for new image prediction, the layer will utilize the trained mean/variance rather than the mini-batch mean/variance for activation normalization.

The three main characteristics are representative of CNN models and indicate their potential for image forgery detection. These characteristics are presented below:

• Convolution operation: This is defined as adding image pixels within local regions, thereby accumulating into large values the duplicate patches in the area. The large-value accumulation could result in easier detection of forged images among pristine ones [37].

• CNN model convolutional: This is a form of exploitation of any strong spatially local correlations which could occur in input images. Embedded copy-move distorts image pixel local correlation, which then differentiates it from correlations of pristine

images via the process of correlation-based alignment. In this way, any distinctions between distorted and natural images are easily perceived through the CNN models [37].

• Nonlinear mappings: In CNN models, this type of mapping enables them to derive deep and rich features that enable them to be used to classify all types of images. Such features are automatically learned via network updates and would be difficult to apply using the traditional non-CNN method [29].

Table 4.2: Summary of CNN layers with their chosen parameters.

Layer	Preperties	No
imageInputLayer	64×64×3	1
convolution2dLayer	64 5×5 convolutions with stride [1 1] and padding [2 2 2 2]	3
MaxPooling2DLayer	Name: HasUnpoolingOutputs: 0 NumOutputs: 1 OutputNames: 'out' Hyperparameters PoolSize: [2 2] Stride: [2 2] PaddingMode: 'manual' PaddingSize: [0 0 0 0] stride [1 1] and padding [2 2 2 2]	3
fullyConnectedLayer(x)	64 fully connected layer 2 fully connected layer	2
ReLU	ReLU	4
Softmax	Softmax	1
C-Outputlayer	64×64×3	1

The literature, as mentioned above, introduced different types of algorithms used for image forgery in general and in copy-move forgery detection in particular. However, CNNs are emerging now as a powerful method to do the job. The CNN pipeline begins the extracting process of the features from the image using the different layers and then feeds them into the specific classifier to detect the copy-move forgery if it exists. However, before we go over the different parameters used in this CNN, we should first clarify why CNNs are generally a more viable option for this task. The fact that CNNs are learnable method makes them a better choice overall compared to other methods for achieving the same goal. In the evaluation section, we show the output performance of this algorithm versus the state-of-the-art. Secondly, the classifier here can work at the feature level as well as the pixel level, which eliminates the challenge of losing pixel interaction if we use a pixel vector. The CNN uses the first convolution layer to downsample the image by adjacent information of the pixels. The convolution is thus a summation of the weight of pixel values in the input image. This is achieved, in the proposed network, by convoluting the input image 64×64 with a 5×5 Kernel filter. The operation (using a weight matrix) will produce a new image with a smaller size. Each convolutional layer in the CNN will produce multi convolutions, thus generating a weight tensor according to the n number of the convolutions, in this case, the tensor will be 5×5×n. The first convolution layer in the CNN will give a weight matrix of 64×5×5, which will produce 1600 parameters. At the end of the network, we use a prediction layer to support the final classification task. For the last two convolutional layers we padded them with 2, however, the max-Pooling Layer has a pool size of 3×3 and a stride of 2×2.

4.3.1 The proposed CNN architecture

Recent studies show that CNNs are performing remarkably well in image forgery [29][32]. Therefore, in this paper, we propose an end-to-end deep learning CNN to handle and detect copy-move forgery. The proposed CNN includes the following main operational layers: an input layer, convolutional layers, fully connected layers, classification layer, and output layer with each convolutional layer including different convolutional filters. The top benefit of using CNNs in copy-move forgery detection model is the strategy's success in feature extraction which improves the model overall performance. Moreover, improvements in the output results are based on CNN learning skills which can be boosted by increasing the input samples and training cycle. CNNs also lower the cost of detecting copy-move forgery compared with the classic method. Finally, a wide range of input images can be used by CNN which, indeed, increases the output accuracy of the model.

In this paper, the CNN structure is intended for copy-move forgery detection. To that end, we layered the CNN in a specific sequence such that it could function as a type of feature extraction system that uses filter sets of a certain size. The filters are arranged in parallel to the input image regions, incorporating an area of overlap known as the stride. Every convolutional filter output per convolutional layer stands for a feature map or learned data representation. The subsequent convolutional layers likewise extract features from maps which were learned from earlier convolutional layers. The proposed CNN will learn how to detect similarities and differences in image features through a number of hidden layers. Each individual hidden layer will enhance the CNN's learning feature ability in order to increase its detection accuracy. Note that, hierarchical feature extractor output is added to an FC to carry out a classification task learning weight which is first randomly initiated and then learned via a backpropagation method [38]. However, the hierarchical convolutional layers

create an enormous amount of feature maps, rendering the CNN's impractical from both cost and computational perspectives [39]. The network shown in Figure 4.3 is applied to the present study features 15 layers in total: one each of input and output classification layers, one SoftMax layer, one max-pooling layer, two average-pooling layers, two FC layers, three convolution layers, and four ReLU layers.

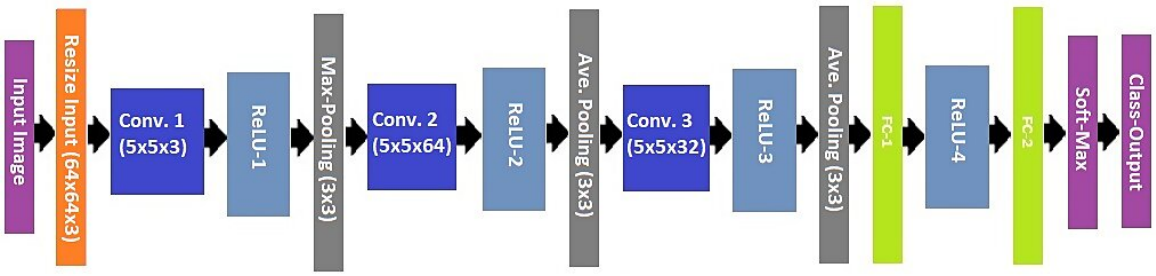

Figure 4.3: The proposed network architecture (The training algorithm of the proposed CNN model

.

4.3.2 Batch normalization of CNN

Batch normalization for CNN has been commonly applied as a technique for classifying output images. Deep neural network model training can be challenging due to data changes across the various different layers (known as the internal covariate shift) as well as gradient vanishing/exploding phenomena [40]. Batch normalization can overcome these issues through the application of a few simple operations for input data, as follows [41]:

$$\mu\rho \longleftarrow \frac{1}{m}\sum_{i=1}^{m} I_i \tag{4.11}$$

$$\mu\rho \longleftarrow \frac{1}{m}\sum_{i=1}^{m}(I_i - \mu\beta)^2 \tag{4.12}$$

$$\widehat{I}_i = \frac{I_i - E[I_i]}{\sqrt{\sigma_\beta^2 + \epsilon}} \tag{4.13}$$

$$I_i^{\circ} = \gamma \widehat{I}_i + \beta \tag{4.14}$$

where I_i indicates the $i - th$ training sample; m denotes batch sample amount; $\beta = I_{i,.....m}$ expresses mini-batch input data; μ_β and σ_β stand for mean and standard deviations, respectively, in mini-batch B; ϵ represents a negligible constant that prevents zero from being divided, and γ and β indicated scale and offset parameters. Against these operational parameters, the mini-batch I_i° output data show a standard deviation and a fixed mean for all depths following normalization of the batch. Hence, any deviations of mean or variance are removed through the process of batch normalization, allowing the network to avoid potential internal covariate shifts.

Different types of CNN's are proposed to achieve a similar goal by employing different architectures and different domains. However, the CNN centered deep learning approach is currently widely used for universal image manipulation and forgery detection [1]. The proposed copy-move forgery detection algorithm is performed based on CNN to adopt an end-to-end structure. Thus, the proposed algorithm provides a better outcome for copy-move forgery detection than traditional copy-move forgery detection algorithms. The copy-move forgery detection baseline initiates by taking the input image, extracting the features, producing feature maps, and then making useful feature statistics with the percentage pooling process of up sample feature maps. After that, the feature classifier can be applied to doctor similar regions as a copy-move forgery. PatchMatch was implemented to achieve the localization assignment.

4.4 Experiment Results

In this section, we present the results analysis and performance of the used CNN deep learning model. Next, we evaluate the model's method versus state-of-the-art approaches. Finally, we present the training experiment and testing evaluation in detail.

4.4.1 Environment Analysis

In this work, we use a CNN deep learning model with two fully connected layers. The auto-resizing layer was modified to inject unrestricted size images and output modified union dataset size to 16×16×3 to fit with the input to the first convolutional layer. Training and testing phases were performed using neural network toolbox-MATLAB 2018a. Learning training was implemented with different image batch sizes: 64, 100 and 265, with the same preliminary learning rate of $10^{(-3)}$. However, the best performance of error loss was accomplished with the mini-batch size of 100. Forgery localization used images with a minimum Size of 277×277. We considered PNG image formats for the used datasets, each image of which is 12.288 k. bytes on the disk versus the actual size of 8.420 k.bytes. A lab machine was used to run this implementation using 16GB RAM. All network parameters were set to achieve smoothed training for both, applying the same number of iterations to test accuracy and loss. In our training and testing, we split the dataset into randomized bases; however, the dataset divided into 70% training data and 30% testing data.

The used dataset is a combination of public online datasets available from research or dataset producers. These publicly available datasets are quite small, however, and none of the existing CMFD datasets provide ground truth masks showing the distinguishing source and target copies. Therefore, we generated a collection dataset out of

online and public existing datasets for training and testing. In total, we collected 1792 paired images of good quality to present different samples for copy-move forgery, each with one binary mask distinguishing source and destination. This dataset contains 166 authentic and 1626 tampered color images. However, in the training task, we don't specify which images are manipulated in a copy-move manner and which are not. Hence we randomly verify that 30% of the total forged samples are a copy-move forgery for testing i.e. around 340 mixed images for testing. These CMFD samples and their authentic counterparts together form the training and testing datasets.

The first one was constructed by Christlein et al [1], consisting of 48 base images and 87 copied with a total of copy-move forged images of 1392. The second database MICC-F600 was introduced by Amerini et al [37][17] with 400 images. The CIFAR-10 (11000) [42]. Caltech-101, image manipulation dataset. IM dataset (240 images) [2]. The Oxford buildings dataset (198 images and 5062 resized images) [42]. Coverage dataset (200) [43] and collection of online and self-producer images. . Note that, even the total images look bigger than what we used in training and testing, that is because we avoid using some images either because they are in bad shape or low resolution. An image data augmentation configures with main properties is shown in the next table. Data augmentation typically will maintain the generalization of the image classification properties such as rotation, scaling, shearing, etc. Training and testing related have been illustrated comprehensively in the result discussion section.

4.4.2 Training

While CNN training involves a larger portion of data, there are no large public datasets that contain numerous image pairs marked with their copy-move manipulations and ground truth. Therefore, we generated our own dataset, from datasets we found online. The training data were designed to present two datasets categories: pristine

Table 4.3: Data augmentation properties

Data	Property	Option value
Input Size	[64 64 3]	Various
Fill Value	0	
Rand X Reflection	0	
Rand Y Reflection	0	
Rand Rotation	[-20 20]	
Rand X Scale	[1 1]	
Rand Y Scale	[1 1]	
Rand X Shear	[0 0]	
Rand X Translation	[-3 3]	
Rand Y Translation	[-3 3]	
Initial Learn Rate	0.01	0.001
Mini Batch Size	256	100.64
lower Threshold	10	8
Validation Frequency	50	30
Base Learning Reta	0.0001	

and forged. The second dataset category is larger than the first because of the different types of geometric transformation employed to the copy-move patches in the forged images. Training images constitute images that have a known outcome. The elements and features of these kinds of images undergo a classification process in order to find their correct weight category. After determining which weights will be used, sample images, whose outcome is also already known, are run. Next, the sampled test images undergo an extraction, while the weight is used to predict image classification. Finally, an actual known classification is compared with the predicted classification to gauge the accuracy of the analysis.

4.4.3 Result and Discussion

In assessing the proposed model approach, we will review the dataset, analyze its performance, and then compare the method to other key algorithms as a reference point. The dataset was built with images that were readily available online. These images were then resized to 64 x 64 and constituted the two specific pre-set image categories of pristine and forged. We used both of these categories for network training, starting with the input layer sized to the output of the automatically resized layer. We also used two learned connected layers – fc1 output at size 64, and fc2 output at size 2. The SoftMax layer represents the final layer used for output discrimination, as shown in Figure 4.3. The variant scale classifier trains the network output at a certain size based on loss function software. The various minibatch sizes used (e.g., 64, 100 and 256) indicate a strong impact on the training set as well as in the saturation of the overall accuracy and error loss. The Figures (4.4, 4.5 and 4.6) show the different training responded based on changing minibatch size and some other important parameters. Moreover, the model fitting shows different training responses based on changes in minibatch size and other important parameters. For instance, the training cycle, for the same data in the same training environment, using minibatch 64, there are 154 iterations. for 7 epochs, i.e., 22 iterations for each epoch. On the other hand, while using minibatch 256, the training cycle will take only 98 iterations for the same number of epochs, but each epoch, in this case, takes 14 iterations to be finished. In both training cases, samples will take roughly the same amount of time. Overall, we found that the best minibatch size is 100. Despite the training process having a high noise ratio, this batch size still gives the best training accuracy and error drops faster, resulting in less error. We then reduce the number of epochs to avoid overfitting during the training task as the input data is for the dataset is not large enough. See Figures 4.4, 4.5 and 4.6.

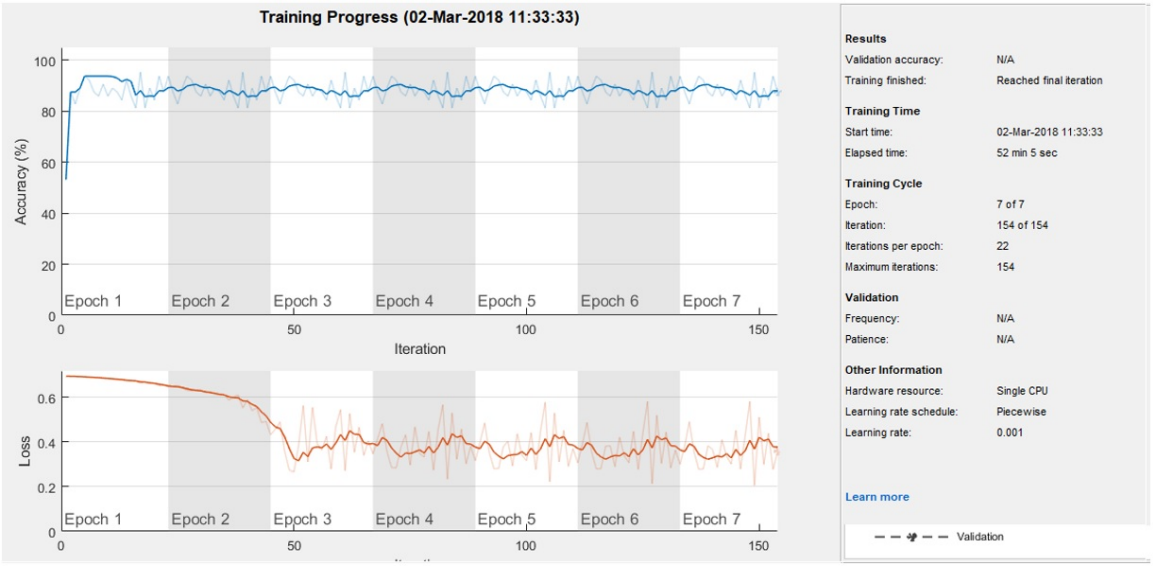

Figure 4.4: Training progress: Mini batch size is 64

.

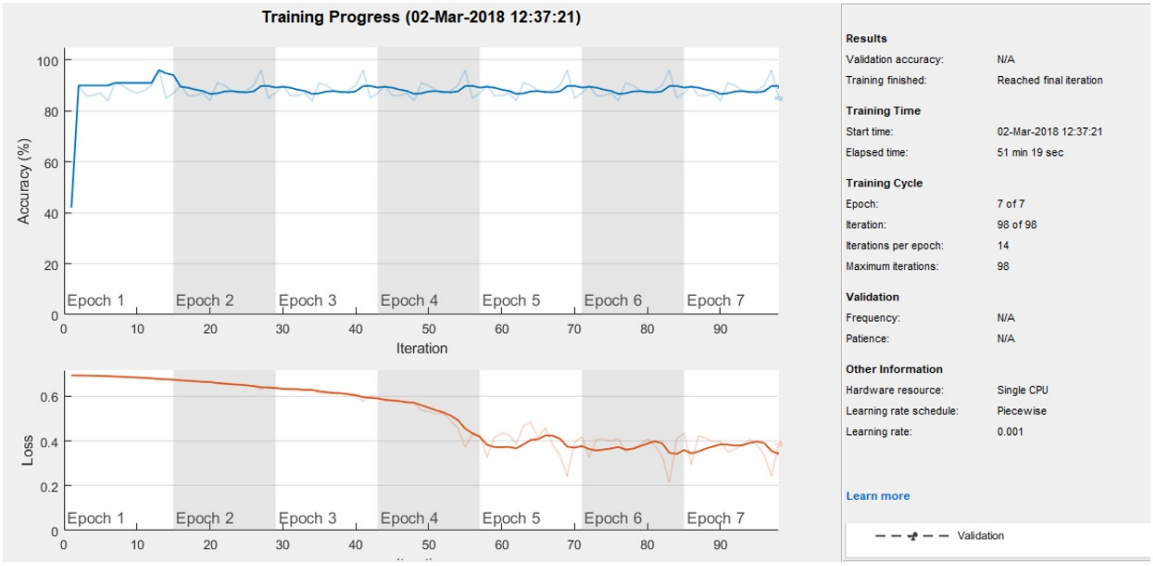

Figure 4.5: Training progress: Mini batch size is 100

.

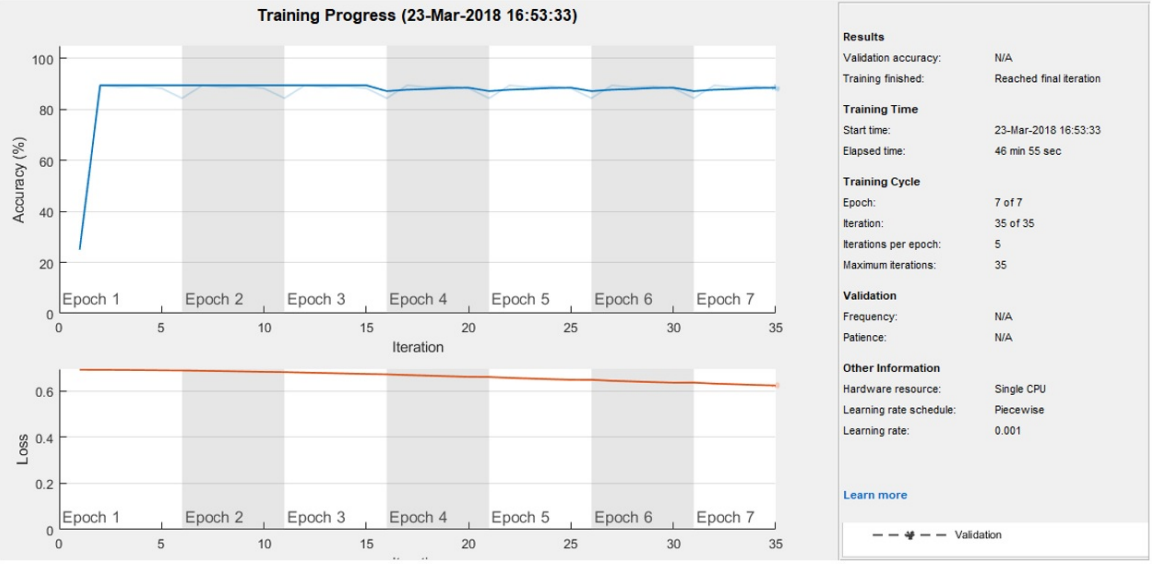

Figure 4.6: Training progress: Mini batch size is 256

.

The results indicate network robustness, despite the small size of our dataset. Given these results, we anticipate that increasing our dataset size will result in even higher efficiency, as the small dataset could not use much of the temporal information i.e. the use of a small sequence of image volumes across the time range will not make an effective investigation to understand the dataset's temporal dynamic. Nonetheless, the good performance of our model still leaves room for improvement, for instance, the approach gives similar results if no post-processing is performed. Overall, the technique provides the best results when applied to active copy-move forgery, whereas for passive copy-move forgery detection, it gives fair results. Figures 4.7 to 4.10 illustrate some results for different scenarios of copy-move forgery detection using the proposed learned CNN approach. As we mentioned in the introduction, CNN still suffers from forgery localization for copy-move since it is located in the same neighborhood as

the original region. Provisionally, we overcome this issue by employing a PachMatch technique to match the feature points between the two regions, with this stage being done separately [44]. However, image patch size used for training and testing, as mentioned above, is customized to small sizes according to network sizing parameters. This will reduce the image size leading to loss of some important details. Hence, this size will not work effectively for forgery localization which mainly relies on the offset points matching. Therefore, the image size used for copy-move region localization is 277×277 instead of the 64×64 used for the training and testing stages.

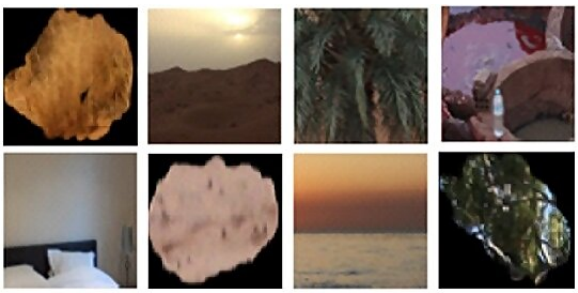

Figure 4.7: Random output samples show the true detection of the pristine images category. The output is flagged with the category name in green color "Pristine" which indicates the correct decision

.

In this Figure 4.7, the model was able to justify the authenticity of these images and mark them as pristine images, which illustrates that the false positive is zero and the true positive is the one in the present experiment. On the other hand, in both Figured 4.8 and 4.9 the model red flags these images as forged images regardless of whether the copy-move forgery type is active (as is in Figure 4.8 or passive (as is in Figure 4.9. These two cases are called true positive and true negative respectively.

An unfortunate scenario occurs when the model marks the forged image as a pristine image. In this case, the model accuracy is reduced. However, even if a result mistakenly shows the image as pristine, the resulting flagging alarm may incorrectly

act by sending the output category in red color as illustrated in Figure 4.10.

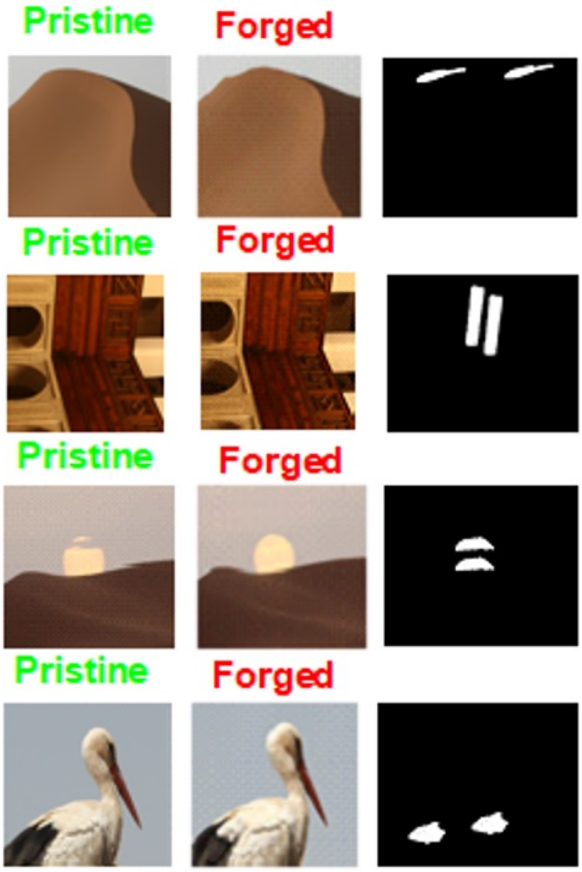

Figure 4.8: This figure shows three image categories (a) pristine image (b) the same image was manipulated with copy-move forgery (c) the output mask shows the copy-move forgery detection result includes the two similar areas in the same image frame.

.

.

.

.

Our model represents a deep learning method suitable for detecting forgery embedded within digital images. Non-deep-learning traditional methods, such as [22], are unable to extract relevant data from input image patches automatically, nor can they devise representations very efficiently. Many non-deep-learning approaches also

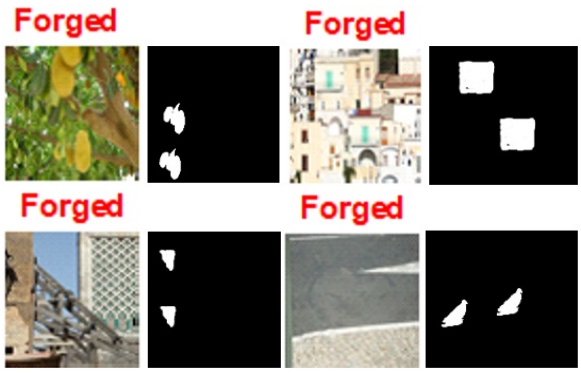

Figure 4.9: Random samples of the model result illustrates a true passive forgery detection i.e. there is no original images (Pristine)

Figure 4.10: False detections by labeling this forged image as a pristine image. There will be a red flag on the result refers to a false result

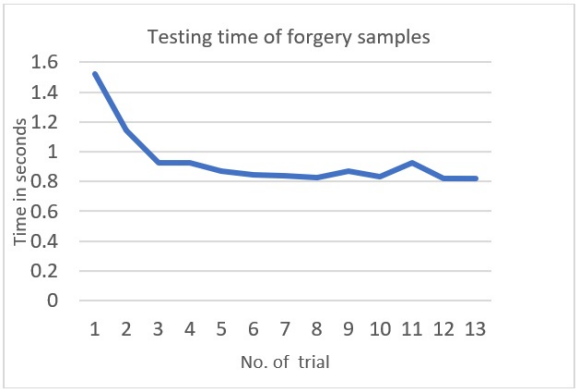

Figure 4.11: Testing time of forged samples related to the number the trials

only utilize a single artificial feature for classification purposes. These are all significant drawbacks in the traditional models. Our proposed method, on the other hand, is much more efficient. It can apply several epochs in the training sets, the optimal number being no less than three epochs and no more than five, which is related to dataset size. Testing a new input image will take a longer time the first time around but will decrease several trails, the first trail case usually takes no more than 1.6 seconds as illustrated in Figure 5.1. This time is based on image resolution and the used machine. Table 4.4 indicates a clear reduction in accuracy from 90% to 81%. The average validation loss rate of the training set was around 0.3010 for all saturated iteration values. Of the 1255 forged images we used, we had an overall validation accuracy of 88.26% to 90.1%. The matrices and the baseline evaluation settings were devised by computing false positive (FP), true positive (TP), false positive (FP) and false negative (FN) settings in order to compute the F-measure. The evaluation scores in Table 4.5 present the F-measure of the proposed model vs the state-of-the-art models [45, 46][28]. It is worth mentioning again that our testing dataset was relatively small and used a mix of both forged and pristine images. Hence, we anticipate that the value will change in accordance with dataset size. Note that the number of epoch is low according to the dataset size to avoid overfitting during the training task.

Table 4.4: The accuracy based the epoch and the number of the iterations

Epoch	Iteration	Time elapsed sec.	Mini-batch accuracy	Base learning rate
1	1	31.42	88.00%	0.0010
4	50	1587.43	90.00%	0.0010
7	90	3120.95	91.00%	0.0010

.

Table 4.5: Comparison of copy-move forgery detection F-measure, precision and recall of different algorithm

Algorithm	[45]	[47]	[48]	[28]	[49]	[50]	Proposed
F1	0.5943	0.5439	0.6055	0.6318	0.7993	0.4926	0.8835
Precision	0.5440	0.5390	0.5662	0.5927	-	0.5734	0.6963
Recall	0.8020	0.8327	0.8040	0.8220	-	0.4939	0.8042

4.5 Conclusion

A novel neural network-based copy-move forgery detection strategy was proposed in this work. The convolutional neural network (CNN) was built with MATLAB due to its ease of use and its support of GPU/CPU computations. Weights for decreasing error rates and improving overall efficiency were applied via backward and forward propagation. Our CNN learned how to reproduce both forged and pristine outputs in its training phase, enabling copied regions to trigger detection during reconstruction. The results of active copy-move detection were highly promising, while the passive detection results were only satisfactory. Additionally, overall efficiency was relatively low due to the small size of the experimental dataset utilized in the training phase. The proposed model's key contribution is its capability of detecting and localizing copy-move forgery. In future related work, other network structures could be tested, and in-depth analyses could be performed through implementing a more expansive dataset than the one used here. As well, other kinds of image manipulation could be incorporated, including post-processing strategies. Additionally, future work could focus on producing customized layers to distinguish the source and target location of the copy-moved region in this type of forgery, or to examine the effects of other shallow learning methods for image copy-move forgery.

4.6 References

[1] Jing, W., Hongbin, Z. "Exposing digital forgeries by detecting traces of image splicing," in 8th International Conference on Signal Processing, vol. 2, IEEE, 2006.

[2] Christlein, V., Riess, C., Jordan, J., Riess, C., Angelopoulou, E. "An evaluation of popular copy-move forgery detection approaches," IEEE Transactions on Information Forensics and Security 7(6), pp. 1841-1854, 2012.

[3] Ryu, S.J., Kirchner, M., Lee, M.J., Lee, H.K, "Rotation invariant localization of duplicated image regions based on Zernike moments," in. IEEE Transactions on Information Forensics and Security, pp. 1355-1370, 2013.

[4] Haodong Li, Weiqi Luo, Xiaoqing Qiu, and Jiwu Huang," IEEE Transactions on Information Forensics and Security, vol. 12, no. 5, pp. 1240– 1252, 2017.

[5] Paweł Korus and Jiwu Huang, "Multi-scale analysis strategies in prnu-based tampering localization," IEEE Translated Transactions on Information Forensics and Security, vol. 12, no. 4, pp. 809–824, 2017.

[6] G. Ulutas, and G. Muzaffer, "A new copy move forgery detection method resistant to object removal with uniform background forgery," Hindawi Mathematical problems in engineering, pp. 1-19, 2016.

[7] Lee, H., Ekanadham, C., Ng, A.Y, "Sparse deep belief net model for visual area," In: Advances in Neural Information Processing Systems, 20 (NIPS), 2008.

[8] Larochelle, H., et al, "Exploring strategies for training deep neural networks," J. Mach. Learn. Res. 10(10), pp. 1–40, 2009.

[9] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P, "Gradient-based learning applied to document recognition," Oroc. IEEE, pp. 2278-2324, 1998.

[10] Giacinto, G., Roli, F., "Design of effective neural network ensembles for image classification purposes," Image Vis. Computer, pp. 699-707, 2001.

[11] Fukushima, K., Miyake, S., Ito, T., "Neocognitron: a neural network model for

a mechanism of visual pattern recognition," IEEE Trans. Syst. Man Cybern, pp. 826–834, 1983.

[12] A. Voulodimos, N. Doulamis, A. D. and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," Hindawi Computer Intelligence and Neuroscience, pp. 1-13, 2018.

[13] Andrea V., K. L., Ankush Gupta, "Convolutional Neural Networks for Matlab," MatConvNet, pp. 1-59, 2015.

[14] He, K., Zhang, X., Ren, S., Sun, J, "Deep residual learning for image recognition," In: arXiv preprint arXiv:1512.03385, 2015.

[15] Krizhevsky, A., Sutskever, I., Hinton, G.E, "Imagenet classification with deep convolutional neural networks," In: Proceedings of the Advances in neural information processing systems, pp. 1097–1105, 2012.

[16] Ren, S., He, K., Girshick, R., Sun, J. "Faster r-cnn: Towards real-time object detection with region proposal networks," In: Proceedings of the Advances in neural information processing systems, pp. 91–99, 2015.

[17] He, K., Zhang, X., Ren, S., Sun, J, "Identity mappings in deep residual networks," In: arXiv preprint arXiv:1603.05027, 2016.

[18] Smeulders, A.W., Worring, M., Santini, S., Gupta, A., Jain, R, "Content-based image retrieval at the end of the early years," IEEE Transactions on Pattern Analysis and Machine Intelligence 22(12), pp.1349–1380, 2000.

[19] Yang, J., Yu, K., Gong, Y., Huang, T, "Linear spatial pyramid matching using sparse coding for image classification," In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 20–25. IEEE, 2009.

[20] Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D, "Object detection with discriminatively trained part-based models," IEEE Transactions on Pattern Analysis and Machine Intelligence 32(9), pp. 1627–1645, 2010.

[21] Jegou, H., Perronnin, F., Douze, M., Sanchez, J., Perez, P., Schmid, C, "Aggregating local image descriptors into compact codes," IEEE Transactions on Pattern Analysis and Machine Intelligence 34(9), pp. 1704-1716, 2012.

[22] Zeiler, M.D., Fergus, R, "Visualizing and understanding convolutional networks," In: Proceedings of the European Conference on Computer Vision, pp. 818–833. Springer, 2015.

[23] Simonyan, K., Zisserman, A, "Very deep convolutional networks for large-scale image recognition," In: arXiv preprint arXiv:1409.155, 2015.

[24] Xie, S., Girshick, R., Dollar, P., Tu, Z., He, K, "Aggregated residual transformations for deep neural networks," In: arXiv preprint arXiv: 1611.05431, 2016.

[25] S. Das, "CNNs Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more," Medium, 2017.

[26] Paulin, M., Douze, M., Harchaoui, Z., Mairal, J., Perronin, F., Schmid, C., "Local convolutional features with unsupervised training for image retrieval," In: Proceedings of the International Conference on Computer Vision, pp. 91–99. IEEE, 2015.

[27] Yaqi L., Q. Guan, X. Zhao, "Copy-move Forgery Detection based on Convoluational Kernel Network," arXiv:1707.01221[cs.CV], pp. 1-26, 2017.

[28] Soni, B., Das, Thounaojam, D. "Cmfd: A detailed review of block based and key feature-based techniques in image copy-move forgery detection." IET Image Processing, 2017.

[29] Birajdar, G.K., Mankar, V.H. "Digital image forgery detection using passive niques." A survey. Digital Investigation 10(3), pp. 226-245, 2013.

[30] Asghar, K., Habib, Z., Hussain, M. "Copy-move and splicing image forgery detection and localization techniques." A review. Australian Journal of Forensic Sciences 49(3), pp. 281-307, 2017.

[31] Y. Huo, X. Zhu, "High dynamic range image forensics using cnn," arXiv: 1902.10938.2019.

[32] P. Isola, J.-Y. Zhu, T. Z., A. A. Efros. "Image-to-image translation with conditional adversarial networks." Conference on Computer Vision and Pattern Recognition, 2017.

[33] L. Bondi, L. Baroffio, D. Guera, P. Bestagini, E. J. Delp, and S. Tubaro, "First Steps Toward Camera Model Identification with Convolutional Neural Networks," IEEE Signal Processing Letters (SPL), 24: pp. 259–263, 2017.

[34] Ioffe, Sergey, and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," preprint, arXiv:1502.03167, 2015.

[35] Ye Yao, Y. Shi, S. Weng, and B. Guan, "Deep Learning for Detection of Object Forgery in Advanced video," Symmetry, pp. 1-10, 2017.

[36] A. Mahendran, and A. Vedaldi, "Visualizing deep convolutional neural networks using natural pre-images," International Journal of Computer Vision, 12(3): pp. 233-255, 2016.

[37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9, 2015.

[38] B. Bayar, M. C. Stamm, "Design principles of convolutional neural networks for multi-media forensics," Society for Imaging Science and Technology, pp. 77–86, 2017.

[39] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," Proceedings of the 32nd International Conference on Machine Learning (ICML), 2015.

[40] Songtao w., S. Z., and Yan L, "A novel convolutional neural network for image steganalysis with shared normalization," IEEE Transaction on multimedia, pp.1- 12, 2017.

[41] James P., Relja A. and Andrew Z. [Online]. Available: http://robots.ox.ac.uk/ vg-

g/data/oxbuildings/. Acc. Jan. 2018.

[42] Im, D.J., Kim, C.D., Jiang, H., Memisevic, R.: Generative adversarial metric, 2016.

[43] Younis A., T. Iqbal, M.S., "Copy-Move Forgery Detection Based on Enhanced Patch-Match," International Journal of Computer Science Issues (14), pp. 1-7, 2017.

[44] Li, J., Li, X., Yang, B., Sun, X." Segmentation-based image copy-move forgery detection scheme," IEEE Transactions on Information Forensics and Security 10, pp. 507–518 (2015).

[45] G. Ulutas, and G. Muzaffer, "A new copy move forgery detection method resistant to object removal with uniform background forgery," Hindawi Mathematical problems in engineering, pp. 1-19, 2016.

[46] Li J, Li X, Yang B, Sun X. "Segmentation-based image copy-move forgery detection scheme". IEEE Trans Inform Forens Secur 10(3). pp. 507–518. 2015.

[47] Ryn, S. J. Lee, M.J., Lee, H.K. "Detection of copy-move forgery using Zernike moments." In: Information hiding. Spider. pp. 51-65, 2010.

[48] Yue Wu, W. Abd-Almageed; and P. Natarajan, "BusterNet: Detection Copy-Move Image Forgery with Source / Target Localization," ECCV, 2018.

Chapter 5

Copy-Move Forgery Detection and Localization Using Generative Adversarial Network and Convolutional Neural-Network

Preface.

A version of this chapter has been published in the Journal of information,MDPI 2019; 11: Isue 9, 10.3390/info10090286, 1-26. I am the primary author. Along with Co-authors, Tariq Iqbal and M. Shehata, I conceptualized the idea. I have conducted the literature search and carried out the data collection then drafted the manuscript and subsequently revised the manuscript, based on the feedback from Co-authors and also peer review process. As Co-authors, Tariq Iqbal and M. Shehata assisted in developing the concept and testing the algorithm, reviewed and corrected the model and results. Also, the Co-author T. Iqbal critically reviewed the content. Contribution was made through the team support to developing and testing the algorithm as well as and re-

viewing the manuscript.

Abstract – The problem of forged images has become a global phenomenon that is spreading mainly through social media. New technologies have provided both the means and the support for this phenomenon, but they are also enabling a targeted response to overcome it. Deep convolution learning algorithms are one such solution. These have been shown to be highly effective in dealing even with image forgery that derived from generative adversarial networks (GANs). In this type of algorithm, the image is altered such that it appears identical to the original and is nearly undetectable to the unaided human eye as a forgery. The present paper investigates copy-move forgery detection using a fusion processing model comprising a deep convolutional model and an adversarial model. Four datasets are used. Our results indicate a significantly high detection accuracy performance (95%) exhibited by the deep learning CNN and discriminator forgery detectors. Consequently, an end-to-end trainable deep neural network approach to forgery detection appears to be the optimal strategy. The network is developed based on two-branch architecture and a fusion module. The two branches are used to localize and identify copy-move forgery regions through CNN and GAN.

Keywords: image forgery; copy-move forgery; CNN; convolutional layer; GAN; neural network training.

5.1 Introduction

The hot topic known as “fake news” is becoming increasingly widespread across social media, which for many people has become their primary news source. Fake news is information that has been altered to represent a specific agenda. To support the production of fake news, images that have been tampered with are often presented with the reports. Fake news production has been enabled in recent years due to two main reasons: first, cost reductions of the required image-producing technology (e.g., cell phones and digital cameras); and second, the widespread accessibility of image-editing software from open-source tools and apps. Anyone with a cell phone or digital camera who has online access to the necessary software can now alter images easily and cheaply, for whatever purpose. At the same time, online access enables the images to be sent across a virtually limitless number of platforms, where they can be further altered through imagery-dedicated software (e.g., Photoshop) using tools such as splicing, painting, or copy-move forgery. Considering how easy it is to create fake images as part of a fake news report, there is a critical need for detection methods that can keep up with the latest technology in fraud production. The integrity of an image can be validated through one or more strategies, either alone or in combination, which tests an image for authenticity [1][2][3]. A popular strategy is copy-move image forgery, which involves copying or cloning an image patch into an identical image. The patches to be copied or clones can be either irregular or in regular form. Copy-move image forgery is increasing in popularity due in large part to its ease of use. Furthermore, because the copied/cloned patch has its source in the original image, photometric characteristics between the original and the forgery are essentially the same, making it that much more difficult for the fake to be detected. Since its recent development at around the turn of the present century [4][5], the primary purpose for copy-move forgery detection (CMFD) has been determining

if the imaging probe in question (otherwise known as the query) features any areas that are cloned, and whether this cloning has been performed through malicious intent. The three main types of copy-move forgeries are plain, affine and complex [6]. The earliest CMFD investigations dealt mostly with plain cloning. Interestingly, in [7], the researchers found that human judgment outsmarted machine learning regarding computer-generated forgeries. This could be caused by the current lack of photorealism found in most computer graphics tools. In response to this flaw, various researchers suggested alternative approaches to analyzing digital imagery. Examples include, in [8], statistics extracted from wavelet decomposition and, in [9], statistics extracted from residual images. As well, researchers focused on noise type and level based on recording devices [10], chromatic aberrations [11], and/or demos-icing filters [12]. Additionally, some researchers looked at color distribution differences [13], whereas others examined the statistical properties found in local edge patches [14]. Currently, deep learning is being successfully applied in a range of applications, as demonstrated in [7][15][16]. Given the increasing difficulty to distinguish between photographic and computer-generated forgeries, the need to develop equally sophisticated detection modes is becoming more and more urgent [17]. The latest incarnations of computer-vision-based image forgeries show a significantly higher degree of photorealism than was previously exhibited [18][19]. Especially compelling is image copy-move forgery (CMF), which changes the features of one image by digitally translating one scene as another. CMF is further enabled by generative adversarial networks (GANs), whose sole purpose has it producing "knock-off" images that are virtually identical to the original ones. The present work proposes a novel algorithm that can detect and localize digital image copy-move forgery. The approach is derived from a new strategy for deep neural architecture that detects CMFs by applying generative adversarial networks. The proposed algorithm assumes that because no forged images

will be accessible for training purposes, feature representations for pristine images are then made available solely on the basis of training tasks. Therefore, a one-class SVM (support vector machine) has been trained in alignment with the characteristics of the images in order to gauge distribution. Any forged images can then be determined according to anomalies found in the distribution patterns.

5.2 Related Works

According to feature extraction and matching schemes, copy-move detection strategies are categorized as one of three different types: block-based or patch methods, keypoint methods, and irregular region-based methods. In the first approach, block-based methods have applied chroma features in some research studies such as [20][21], as well as in PCA features [22], blur moments [23], DCT [24], and Zernike moments [25]. However, this is a relatively computationally expensive approach compared to the other two. The second approach category – keypoint-based methods – includes triangles [26], ORB [27], SURF [27[28][29], and SIFT [30][31]. Keypoint approaches are known to be generally fast, but they can fail if S and D show as homogeneous. Meanwhile, the third category deals with strategies related to irregular region-based methods, as in [32] and [33], and has been shown to be somewhat efficient, though occasionally resulting in false positives [34-37]. The latest research on image forgery detection focuses on deep neural networks (DNNs). In [67], DNN was applied for extraction of features in CMF, while in [38], altered areas of the image were detected using a DNN-based patch classifier. The researchers in [39] looked for a way to detect localization and splicing using a DNN solution, and [40] explored how DNN can be used in detecting images that have been doctored. In general, GAN-based methods provide the most optimal results in image forgery detection, with the majority of

the applications using numerous paired images from both domains to train the networks. In instances where no image pairs have been made available for the training process, an alternative method can be used to continue in the GANs approach. The adversarial training paradigm features two main components: the generator (i.e., the image-to-image network) and the discriminator (i.e., the support network). Within the paradigm, the generator's training revolves around learning to deceive the discriminator, while the discriminator is trained to detect real images from forged ones [41]. In [42], Zhu et al. devised a method for automatically pairing images, thus overcoming the shortage in genuine image pairs. For a baseline, Zhu and colleagues [42] employed a discriminator in the GAN. Nonetheless, there are a few methods where explicit delineation of a probability distribution is not done; in these cases, generative machines are trained to obtain samples within a specific distribution source. The main benefit of using this technique is being able to design the machines for the desired training task.

5.3 Proposed Copy-Move Forgery Detection Strategy

5.3.1 GANs create forged images

Advances in technology are enabling GANs to create forged images that fool even the most sophisticated detectors. Keep in mind that the primary aim of generative adversarial networks is to form images that cannot be distinguished from the original source image. Figure 5.1 below depicts image forgery translation enabled by GANs.

.

As can be seen, generator GA has been employed to transform input image A from

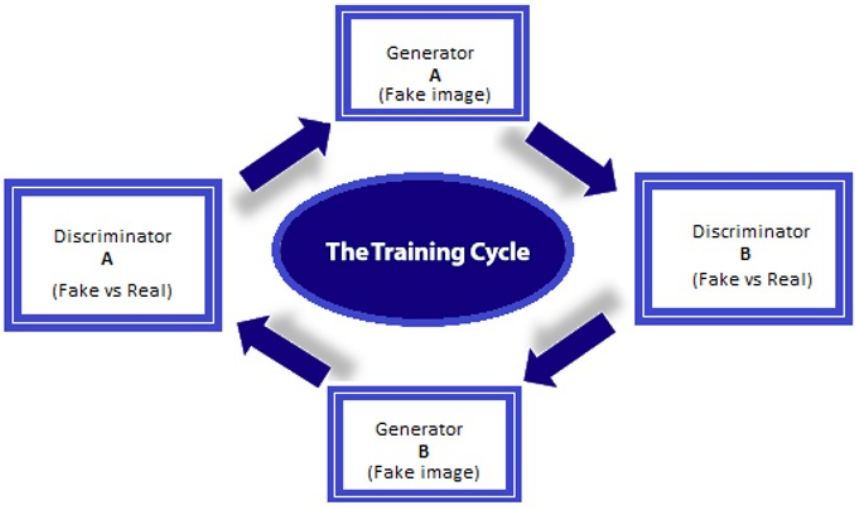

Figure 5.1: The GAN training cycle for fake image translation (i e fake image creation based on real image)

domain DA to output domain DB. Next, generator GB is used to map image B back to domain DA (the original domain). In so doing, two more cycle consistency losses are added to the typical adversarial losses borne by the discriminators, thus obtaining A = GA(GB(A)) and enabling the two images to be paired. Extremely sophisticated editing tools are required to change an image's context. These tools must be able to alter images while retaining the original source's perspective, shadowing, and so on. Those without forgery detection training would likely be unable to distinguish the original from an image forged using this method, which means that it is a good candidate for developing supporting materials for fake news reports.

5.3.1.1 GAN tasks

1- Loading dataset, 2- Building discriminator network, 3- Building generator network, 4- Generating a sample image, 5- Training difficulties, 7- Closing thoughts. Shown in figure 5.2.

5.3.1.2 GAN processing Steps

In our GAN proposed network, we consider three main steps: 1- In the first step, the Generator creates an image from random input. 2- The image is then presented to the Discriminator, together with several images derived from the same dataset. 3- After the Discriminator is presented with the real and forged images, it provides probabilities in the form of a number between 0 and 1, inclusive. Here, 0 indicates a forged image and 1 indicates a high likelihood for authenticity. Note that the Discriminator should be pre-trained prior to the Generator, as this creates a clearer gradient. It is important to retain constant values for both the Discriminator and Generator when training their opposite (i.e., the values for Discriminator when training Generator and vice versa are steady). Holding the values constant enables the networks to have a greater understanding of the gradient, which is the source of its learning. However, because GANs have been developed as a type of game played between opposing networks, maintaining their balance can be challenging. Unfortunately, learning is difficult for GANs if the Discriminator or Generator is too adept because GANs generally require a lengthy training period. So, for instance, a GAN could take a few hours for a single GPU, while for a single CPU, a GAN could require several days [43].

5.3.1.3 Support vector machines

Recently, there has been a decline in reliance on support vector machines (SVM), particularly kernel SVMs, as they require real valued vectors. Users and researchers are instead turning to machine learning systems as end-to-end learning models. In general, deep learning architectures usually source the classifier function and feature representations solely from training examples, but we employed a linear SVM toward the end of every deep convolutional neural network branch. We then trained

them jointly by applying a backpropagation algorithm and stochastic gradient descent (input→→convNet→→SVM→→output). Note that the linear SVM has been given a linear activation function (similar to a regression function), with the hinge loss replacing the loss function, as follows:

$$L(\widehat{y}_i, y_i) = max(0.1 - y_i\widehat{y}_i) \tag{5.1}$$

Where $y_i \in [-1, 1]$ and $\widehat{y}_i = actualoutput$. It is also possible to use a stochastic sub-gradient descent rather than an SGD, given that the hinge-loss cannot be differentiable. The training thus occurs end-to-end, with the hinge-loss error signal guiding the convNet and SVM weights learning. Furthermore, because hinge-loss causes the linear units to connect with learning maximum margin hyperplanes, linear SVMs are the outcome. As an example, a linear unit is connected by the hinge-loss, resulting in a single linear SVM + a trained convNet as feature detection after training. In this setup, the SVM is designed to learn the final splitting hyperplane and the convNet is designed to learn the hierarchical features. Following the training procedure, a heaviside step function can then be applied against the linear SVM output to obtain a binary output. The fact of using SVM in forgery detection is based on capturing the difference before knowing for certain that the patch is a forged which requires the use of a one-class SVM. This entity, which has been trained using feature vectors h that have been extracted from input images, quickly learns pristine feature distribution versus the forgery feature distribution. Next, it outputs a soft value that indicates the degree of possibility that the feature vector h is pristine or forged. The soft mask $M^{'}$ is then defined as a matrix which has identical dimensions as the image, with every entry having a soft SVM output corresponding to image patches at identical positions. A final detection binary mask M can be obtained by employing the soft mask $M^{'}$ for thresholding. More details will be provided in the coming section.

A summary of the copy-move forgery detection (CMFD) strategies employed in the present work is given in this section. Figure 5.2 illustrates the pipeline for the proposed technique. As shown in the figure, our proposed approach employs two distinct networks for forgery detection. One of the networks is GAN-based detects any symptom of forgery, while the other locates any similarities existing in the image. As a joint network, the proposed method then locates any copy-move forgery found in the imagery target, demarcating the original source from the copy-moved region of the image. In CMFD related task, the input data proceeds through the two networks GAN and CNN, and then assigned to a linear classifier based on the proposed model selection. The CNN in general maintained a feature extraction and ability to generate features versus strong discrimination skills, therefore proposed model is used it for data generation, feature extraction, data discrimination, and data classification.

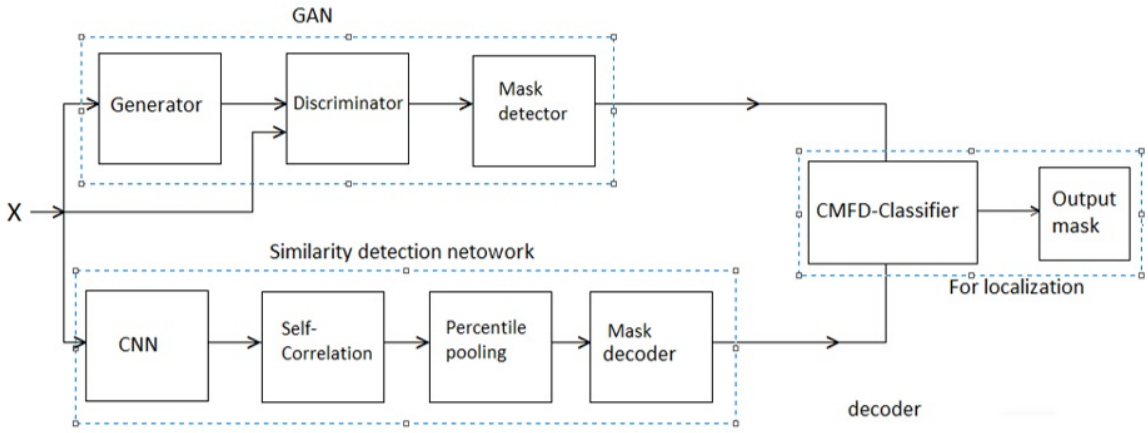

Figure 5.2: A layout of the proposed model

.

5.3.2 CNN in matching or detecting similar patches

A critical issue in copy-move forgery detection is images which contain features that are nearly identical, although matching visual content for the same or diverse im-

ages can be done [44]. The important element in matching methods is how much rigidity ensues after the correspondences are computed. In most instances, matching size options are relatively manageable, but they can also be extreme, in which case the problem is unconstrained. Recently, some methods are being developed which can detect matching objects throughout an image and across several different viewpoints [45][46], but improvements can still be made [47]. Such improvements could include involving convolutional deep neural networks, as these networks are relatively easily trainable end-to-end. For local feature representations, researchers have used deep learning in different copy-move forgery detection stages, such as metric learning, descriptor, and detector. Based on preceding research inquiries, the present work proposes employing deep convolutional neural networks (CNNs) to deal with learning automatic detector in similar features presentation [48]. CNNs are especially valuable for learning decision features adaptively when working from large data sets [49]. In the present study, the proposed model will learn a number of attributes, such as what is considered good features, how to capture similar pixels across different scales, and finding possible similarities between patches. The proposed model is then evaluated qualitatively as well as quantitatively, showing its ability to discern and match similar patches across multiple scales. The primary contribution of this work is developing a learning algorithm that can detect copy-move features according to similarities found in detected features within an image frame.

5.3.2.1 Similarity-matching tasks

In similarity-matching tasks, similarities are computed by employing a multi-layer CNN that deconstructs the targeted patches into sub-patches. Under this setup, different scales can be applied and repetitive textures used. Local similarities are computed in every individual layer by starting with the assumption that the feasible

rigid deformations comprise only a limited set. Detection of the matching features is propagated throughout the sub-patch hierarchy, with the incorrect detections being discarded during the process [46]. As can be seen in Figure 5.2, the architecture resembles a traditional computer vision pipeline, except for the application of differentiable modules. The addition of these modules enables end-to-end training to occur in the localization tasks as well as in the main target task of CMFD. The similarity detection network moves all the images (pristine and forged) through the convolutional layers, in the process extracting feature maps by employing CNN. This procedure continues through dense local descriptors and percentile pooling to feature maps, and then, using the mask decoder feature, on to the original image size. At this stage, similar feature maps are matched as a tentative correspondence map, using binary classifiers. Further details will be provided in the training section below.

5.3.2.2 Feature extraction

Traditional CNN architecture is employed for the pipeline's initial step of feature extraction. In this stage, a CNN which does not have fully connected layers develops a feature map from an input image ($f \in R^{h \times w \times d}$). This can also be expressed as a $h \times w$ dense spatial grid from d-dimensional local descriptors. In [35], researchers applied more or less the same interpretation for instance-retrieval, showing that CNN-based descriptors had significant discriminative power. Therefore, as feature extraction in the present work, the VGG-16 network will be employed. This setup features four layers (16 convolutional layers) and conducts $3X33 \times 33X3$ convolutions and $2X22 \times 22X2$ pooling throughout the extraction process. Note that this strategy is presently the most popular for image feature extraction. In employing the technique, however, we decided to modify the terms slightly by cropping the pool4 layer (in front of the ReLU unit) prior to applying per-feature L2-normalization. A pre-trained

model was used to perform image analyses and classification. Figure 5.2 illustrates the duplication of the feature extraction network, showing how it is organized as a series configuration. In this arrangement, the input images move along the network path [22][32]. The aim of conducting feature extraction is mainly to boost the learned models' accuracy through the extraction of the most critical features and the removal of redundancies and noise [25]. Note that, in general, feature extraction focuses on extracting useful information out of raw pixel values; the information is considered "useful" if it can distinguish among various categories. Following successful feature extraction, the images and related labels are then used to train a classification module which will be used in measuring distances toward the detection of similarities.

Table 5.1: Comparison of computational complexity

Cited	Feature Methods	Feature length
Fridrich et al. [50]	DCT	64
Bayram et al. [51]	FMT	45
Popescu and Farid [52]	PCA	32
Huang et al. [53]	Improved DCT	16
Proposed technique	GAN and CNN	16
Toqeer et al. [54]	DCT and KPCA	10

The comparison between the proposed method and other state-of-the-art methods in terms of feature vector dimensions which presented in table 5.1 shows that our method technique uses low feature vector dimension which, in fact, increases the model efficiency in an effective syntactic computational scheme.

5.3.2.3 Feature extraction problems

Accurate and efficient feature extraction of input data is crucially significant in machine learning. In feature extraction, input data are transformed into feature vectors, which in turn become input in learning algorithms. To address the many issues that have arisen over the years in pursuit of optimal feature extraction, researchers have developed a number of techniques, ranging from feature selection to dimensionality reduction to manifold and representation learning [28]. The most promising solution appears to be incorporating CNNs in the VGG16 approach. In this architecture, multiple 3 X 3 kernel-sized filters are used in succession. Multiple stacked small-size kernels function more optimally than large-size ones, as multiple non-linear layers add more depth to the network, allowing for more complex learning and reduced costs. The extraction of relevant features can be done effectively even from a simple approach, and such an approach need not be complex or large to be effective [29].

5.4 Proposed Algorithm Overview

The present study proposes the construction of a copy-move forgery detection algorithm that features two deep neural networks – a GAN network and a custom CNN-based one. The details of the proposed network are as follows. The GAN network contains both the generator and the discriminator. It will be built first, followed by the custom CNN. The third step in the construction involves merging the two output networks to create a merging network branch. In this section, we will sketch down these networks with draining more related details.

5.4.1 Discriminator network

Constructing the discriminator network requires some prior work, beginning with defining the functions to build the CNNs in Tensorflow (e.g., looping and conv. layers). For the Tensorflow CNN, the classifier is explained as follows: https://www.tensorflow.org/tutorials/ Based on this architecture, our discriminator will have many layers, six conv. layers, followed by seven ReLU layers and two fully connected ones. The main purpose of the discriminator is to discriminate the accuracy value between the real patches in the real input image and the regenerated patches in the fake image out of the generator. The working manner and training process of the discriminator will be fully presented in the implementation section.

5.4.2 Generator network

The generator module in figure 5.3 we will use in our proposed construct resembles a reverse-order ConvNet and CNNs, which aims to change into single probability 2D or 3D pixel value matrices. In contrast, generators aim to change d-dimensional input (noise) vectors into 28 X 28 images by upsampling. The generator and discriminator underlying structures are quite similar, but here we will refer to the convolution transpose method rather than the conv2d method.

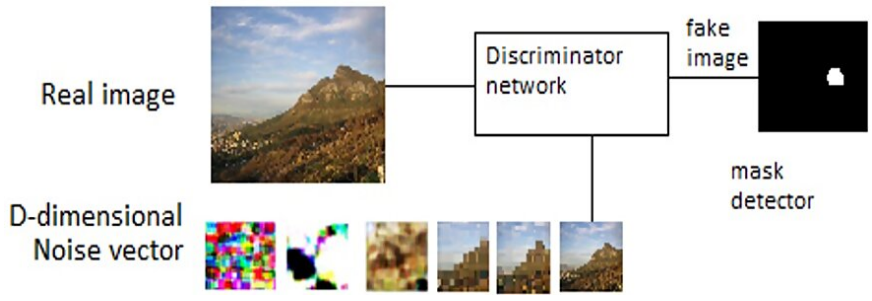

Figure 5.3: Shows the GAN module used in the proposed model

.

Sample image generation: Sample outputs can be generated by untrained generators. Using Tensorflow, we first define the session and assign an input placeholder which we will refine at a later stage. Note that the loss function in GANs can be quite complex compared to standard CNN classifiers. Meanwhile, the generator constantly improves its output images, and the discriminator works to better discern "real" from "generated" images. Hence, the loss functions need to be formulated to take both networks into account. So, discriminator prediction probabilities of real dataset images will be held by D_x; generated images will be held by G_z; and discriminator prediction probabilities of the generated images will be held by D_g. Our goal is to generate images on the generator network which the discriminator will not be able to identify as forgeries. To achieve that end, we start by computing label-of-1 and D_g losses using the function $\imath tf.nn.sigmoid_cross_entropy_with_logits\jmath$. After obtaining the two loss functions (d_loss and g_loss), our next step is defining the optimizers. In the generator network, the optimizer is used for updating the generator's weights only, not the discriminator's weights. Therefore, we need to ensure this distinction, which we can do by devising two separate lists of the generator's weights and the discriminator's weights. After specifying the two optimizers, the better SGD one appears to be Adam. Finally, to update our generator and get a probability score, a random z vector will be fed to the generator and the output passed to the discriminator. Discriminator updates will be obtained the same way, substituting the discriminator for the generator.

5.4.3 CNN networks

CNNs function as feature extractors. We will first use Tensorflow to create the CNNs for self-correlation, looping and conv. Layers as in figure 5.4. These will feed into the mask detector, which will highlight similarities existing throughout the target area

of the image. Next, the extracted features will be subjected to self-correlation, with the module detecting any extracted features which are alike. The percentile pooling layer will then compile the relevant statistics. Finally, the mask detector will be used to recast the feature to its original image size, after which the linear classifier will be applied and a decision made regarding the authenticity of the image.

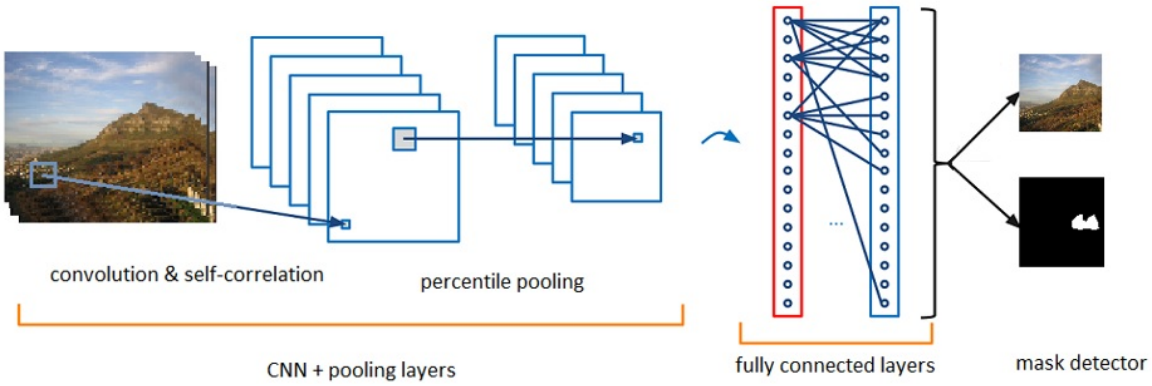

Figure 5.4: Presents the branch used for similarity detection based on CNN

.

5.4.4 Merging network

The outputs of these two networks serve as inputs for the merging one. The three core aims for building this new network are: first, to render a final decision on the copy-move forgery image under study; second, to make the copy-move transaction localized; and third, to make a distinction between the original source and the targeted regions of a potentially forged image. Though varied in their application, neural networks (NNs) are generally used for predicting categorical variables. A typical NN classifier features n input nodes, with n indicating how many values can be assumed by the dependent variable. In the present network, we use as input values the two vectors from the other networks' output. Thus, CMFD classifier output, in this case, is a node which represents the concatenated sum of all relevant inputs. To begin, we

tested all of the model networks individually. At the point where the desired outcome was obtained by the two networks, we used GAN for detecting any forged area(s) in the image and the similarity CNN for detecting any similar area(s). Next, the two outputs were combined into a single layer to represent novel vector inputs, with the first layer being an SVM classifier. To preclude high extremes of randomness, every test was performed multiple times and involved random selection of both the test sets and the training, after which we averaged the results. Terms of expected score defined as

$$S = \frac{P_r(\widehat{F}|F) + P_r(\widehat{P}|P)}{2} \tag{5.2}$$

where $P[F]$ demonstrates the case "image pristine[fake]", $\widehat{P}[\widehat{F}]$ is the case "decision pristine [fake]" and P_r indicates the predicted scores from each case [55]. a- CMFD classifier: we used an SVM linear classifier. Eventually, our SVM classifier uses the merging of the vector features of the two models and is trained over the whole training set. b- Output Masking: shows three images with copy-move forgeries, the corresponding ground truth, and the detection map output by our method. Note that the forgery is easily detected, and the map is quite accurate, although the original and copied regions are distinguished from one another. With an eye to making the model more robust, alternative performance measures were attempted. So, for example, in every SVM classifier, the separating hyperplane was shifted to an orthogonal direction and the subsequent ROC constructed. Next, we calculated the area under (the receiver operating) curve (AUC) for every model, as a sizeable AUC usually indicates robustness, even when functioning under changeable conditions [55].

.

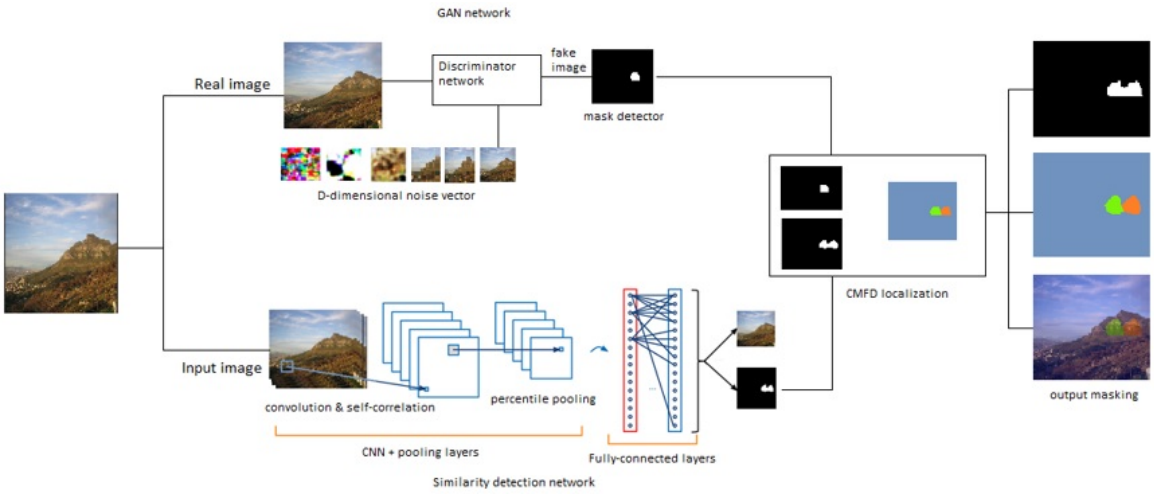

Figure 5.5: Overview of the proposed model two branch DNN based (GAN and CNN) ended by merging network to provide CMFD

5.5 Proposal Implementation

This section provides details on the present study's localization and splicing detection methods; for further details on GAN, please refer to [56]. The GAN used in this work is called CapsuleGAN which trained with forged and pristine images as a means to map input image I in relation to forgery mask M. As shown in Figure 5.2, the GAN architecture comprises a generator G and a discriminator D, with generator G consisting of a 16-layer U-net format of 8 encoder and 8 decoder layers [68][57]. So, for instance, if G receives image I, G will calculate a forgery mask $M^{'}$ (soft mask) as $M^{'} = G(I)$. In so doing, the generator aims to construct an $M^{'}$ as similar as possible to the genuine M. The discriminator D will then differentiate between the generator's constructs (i.e., synthesized input-mask pairs $I, M^{'}$ vs. genuine input-mask pairs I, M). Equation 4 expresses how the discriminator and generator are coupled through a loss function, using cGAN. The training of the generator includes forcing it to construct masks which are so close to the original that the discriminator is unable to distinguish the original from the forged. In this way, the generator is forced

to become better and better at creating near-identical images. The architecture of the discriminator D is a 6-layer CNN which can perform binary classification for masks. Whether an image-mask estimate pair $I, M^{'}$ or a genuine image-mask pair I, M is given to the discriminator, it will subdivide the provided input as pixel patches. Every patch within the subdivision will then be classified either as pristine or forged by assigning the patch label of either 1 or 0, respectively, after which the values of every patch combined will be averaged in order to classify the complete input as a whole. We can use these equations to illustrate the two cases from the paragraph.

$$D(I, M^{'}) = D(I, G(I)) \tag{5.3}$$

$$D(I, M) = 1 \tag{5.4}$$

Essentially what we create is a minimax game, with generator G and discriminator D training through the competition to enhance the skills of the other. The coupled loss function of the network is described in the following equations:

$$L_{cGAN}(G, D) = E_{I,M}[log(D(I, M))] + E_{I,Z}[log(1 - D(I, G(I))] \tag{5.5}$$

Above, we explained how generator G is forced to construct masks $M^{'}$ that can fool the discriminator D, but this process stops short at guaranteeing the synthesized mask can detect an image forgery. So, for instance, if $M^{'}$ can trick the discriminator D into believing it is not a forgery, it will be classified as authentic even though it is not, and $M^{'} \neq M$. To overcome this failure, we can then add another limitation to the generator G, thus enabling it to reconstruct the genuine masks from the original training images, such that $M^{'} \approx M$. We can accomplish this through retraining G and teaching it how to minimize reconstruction losses L_R related to $M^{'}$ and M. Recall that our overall aim is still to categorize each pixel as either pristine or forged, so L_R

is chosen as our binary cross-entropy (BCE) loss. Hence, the total loss function for cGAN can be expressed as follows:

$$L = L_{cGAN} + \tau L_R \tag{5.6}$$

After training has been accomplished, generator G will be able to construct masks $M^{'}$ which are very close to M. Then, for evaluating forgery detection, we can estimate the mean pixel value for a mask as follows:

$$M^{'}_{avg} = \frac{1}{X.Y} \sum_{x=1}^{x} \sum_{y=1}^{y} M^{'}(x, y) \tag{5.7}$$

Where as $X \times Y$ is the image resolution. Binary thresholding can then be used in tandem with threshold T to decide if a specific image I has been forged or not. Images are designated pristine (i.e., not forged) if $M^{'} \approx 0$, or according to thresholding, if $M^{'}\ avg. < T$. If this is not the case, 1 is considered a forgery. We illustrate receiver operating characteristic (ROC) curves showing various threshold T performance levels. Additionally, the ROC will indicates model performance when employing BCE loss and L_1 as reconstruction loss and illustrates that the area under the curve (AUC) designates perfect detection accuracy. The results can be verified against the precision recall (PR) plot for the model, which also shows an excellent detection rate for this work experiment.

Finding the similarity using CNN is just street forward as mentioned above, except including customized layers to do Self-correlation and Pooling procedure in percentile scheme by preference the vector' scores in percentile ranking. Each extracted feature by CNN will produce a feature tensor f_s sized in $16 \times 16 \times 512$. The goal here is to match any similar features by correlating all feature together by applying a self-correlation layer to sorted out the tensor $S[i]$ as follows:

$$S[i] = [\rho_{(i,0)}, ..., \rho_{(i,j)}, ...\rho_{(i,255)}] \tag{5.8}$$

Where $\rho_{(i,j)}$ is the Pearson Correlation coefficient which, in fact, maintains the feature similarity. For instance, here we have two suspected feature patches $f_{(i_r,i_c)}([i])$ and $f_{(j_r,j_c)}([j])$, which can be normalized in the form of $\widehat{f}_{(i_r,i_c)}([i])$ and $\widehat{f}_{(i_r,i_c)}([j])$. The $\rho_{(i,j)}$ to these designated features will be given as follows:

$$\widehat{f}_{(i_r,i_c)}([i]) = \frac{\sum_i (f_{(i_r,i_c)}[i] - \mu[j])}{\sigma[i]} \tag{5.9}$$

$$\widehat{f}_{(i_r,i_c)}([j]) = \frac{\sum_j (f_{(i_r,i_c)}[j] - \mu[j])}{\sigma[j]} \tag{5.10}$$

$$\rho_{(i,j)} = (\widehat{f}_{(i_r,i_c)}([i]))^T \widehat{f}_{(i_r,i_c)}([j])/512 \tag{5.11}$$

Here, $\mu[.]$ is the mean value of the feature $f_{(i_r,i_c)}[.]$ where $\sigma[.]$ is the standard deviation to the same feature and the feature size is defined by: $r, c \in 0, \ldots, 15$. The feature tensor $S[i]$ now can be sorted out by applying Percentile Pooling to new vector called $s^{'}[i]$. By plotting this vector will give a curve shape. The matched features $f_{(i_r,i_c)}[i]$ and $f_{(i_r,i_c)}[j]$ will cause the curve to drop abruptly whenever they exist. The vector' scores in percentile ranking will fix the issue in pooling layer input size by normalizing the score vector. This can be done by applying a percentile ranking filter through Percentile pooling layer as in figure 5.5. This, indeed, will reduce the scores dimensionality to allow using a smaller vector of the total scores. The produced mask will follow same used technique is in the above network by using the same binary classifier. The final step is determined the source patch against the target one by using the merging network.

5.6 Results and Discussion

5.6.1 Training GAN models in forgery detection

The present study proposes using a generative adversarial network (GAN) for a framework to include relevant entities. For instance, cNets have been employed as discriminators, compared to its counterparts GANs and both of them are quantitatively and qualitatively [58]. Our test results indicate a more robust performance by cGANs compared to CNN-based GANs for constructing a model that reflects how CIFAR-10 [69] and MNIST [59] datasets apply the generative adversarial metric (GAM) either quantitatively or qualitatively [60]. As discussed earlier, Goodfellow et al. [61] debuted the GANs framework in order to build a generative model for data that would learn how to transform points from simple prior distribution ($z\ P_z$) to data distribution ($x\ P_x$) using an adversarial generator and discriminator. The generator's task is to learn to transform $G(z)$, while the discriminator's is to goad the generator into performing better $D(.)$. Ultimately, the discriminator must be able to distinguish between a sample from the generator's output distribution ($G(z)\ P_g$) and one derived from data distribution ($x\ P_x$), giving a scalar output ($Y \in 0, 1$).

$$min_G max_D V(D,G) = E_{x\ P_x(x)}[logD(x)] + E_{Z\ P_Z(Z)}[log(1 - D(x))] \tag{5.12}$$

Capsule Networks: Hinton et al. [62][58] first introduced capsules as a learning approach to robust unsupervised image representation. Capsules can be generally defined as locally invariant neuron groups learning to detect visual entities in their midst and encode the properties of those entities as vector outputs. In this process, the vector length is restricted to being either 1 or 0 as an entity representation.

So, for instance, the capsules are able to learn how to discern in images of specific objects or parts thereof. The neural network framework allows for the grouping of numerous capsules, creating capsule layers. In these layers, individual units generate vector output rather than produce traditional scalar activation. Equation 5.2 depicts margin loss L_m when training CapsNets to perform multi-class classifications:

$$L_M = \sum_{k=1}^{k} T_k max(0, m^+ - \|V_k\|^2 + \lambda(1 - T_k) max(0, \|V_k\| - m^-)^2 \tag{5.13}$$

Where T_k indicates target labels, $m^+ = 0.9$, $m^- = 0.1$ and $\lambda = 0.5$, which represent down-weighting factors that can inhibit the shrinking of capsule output lengths in the final layer during the early-stage learning phase. Included in the network as well is regularization, which appears as weighted image reconstruction loss. In this addition, the vector output V_k from the final layers are manifested to the reconstruction network as inputs.

5.6.1.1 Data environment

Our experimental results are provided through several datasets. As a training task, we chose the datasets CIFAR-10 and MNIST. The CIFAR-10 dataset shows 32 X 32 color images categorized as ten distinct classes (in alphabetical order: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck), while the MNIST dataset shows 28 X 28 hand-written grayscale images. When we test the model, additional datasets from both the forged and pristine images categories will be considered. Figure 5.6 depicts the random result samples of the proposed model.

.

As shown, the model is pre-trained to ensure it has the weights to perform both

Figure 5.6: From left to right, some samples from random results show the process of training improving transition result of the model using CIFAR-10 dataset and maximum iteration of 10,000

generator and discriminator training tasks. The pre-training method for the discriminator is illustrated in the next figure 5.7. As indicated, the pre-training of the generator and discriminator are not carried out simultaneously (Pre. Disc, Pre. Gene) = (True/False) or = (False/True). Note that although we chose to use the model's pre-training weights to train the same dataset as well as to test the same dataset at a later time, it is possible that using alternative pre-training model weights could give the same or similar results.

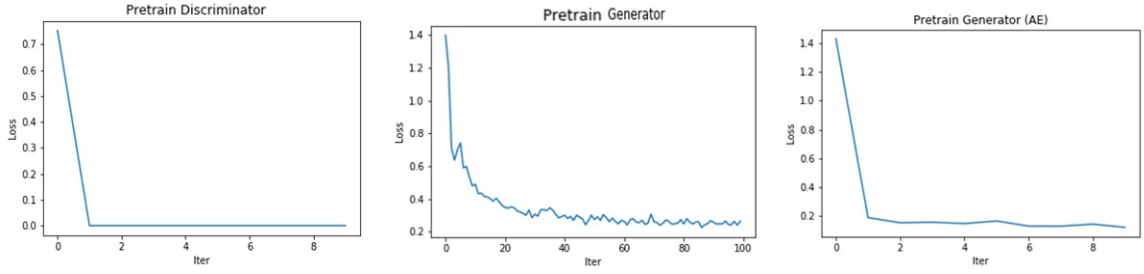

Figure 5.7: Loss function of the pretraining model

.

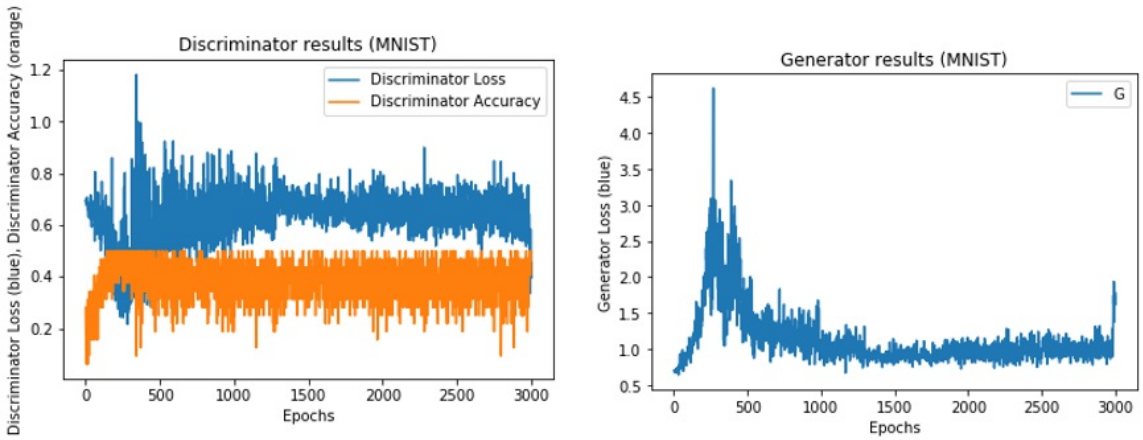

Figure 5.8: This is the training results of MNIST dataset in number of 1000 epochs

.

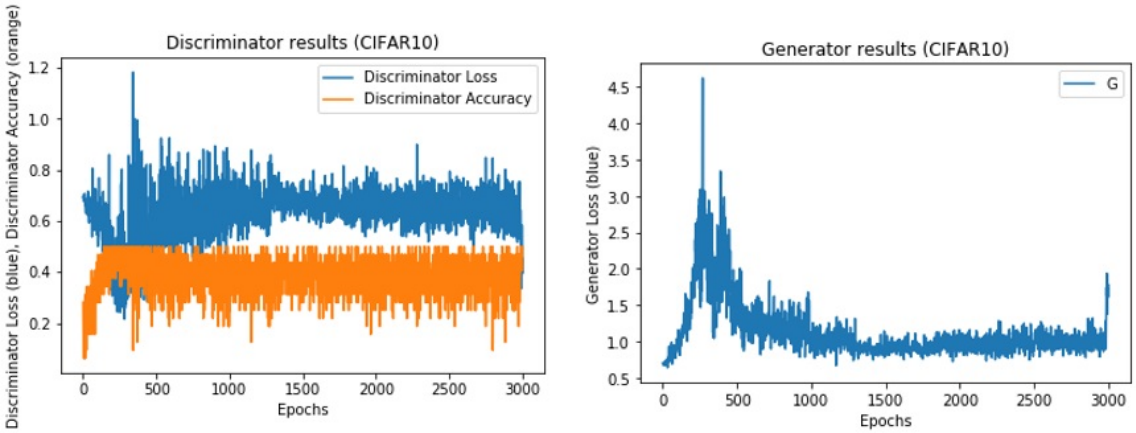

Figure 5.9: This is the training results of CIFAR10 dataset in number of 1000 epochs

.

.

After training the GAN, we test the discriminator by using MNIST: We got these: (D_x,D_z,D_g) as in figures 5.11, 5.12.

.

.

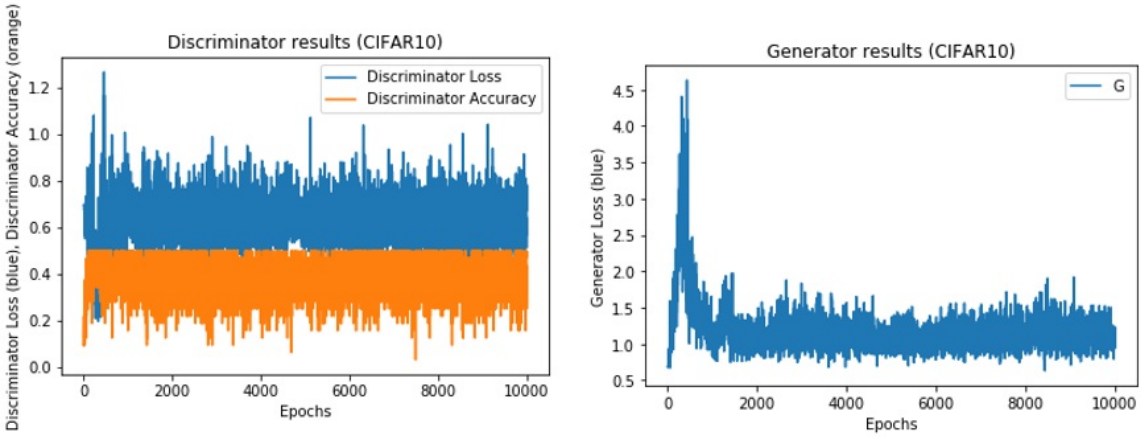

Figure 5.10: This is the training results of CIFAR10 dataset in number of 10000 epochs

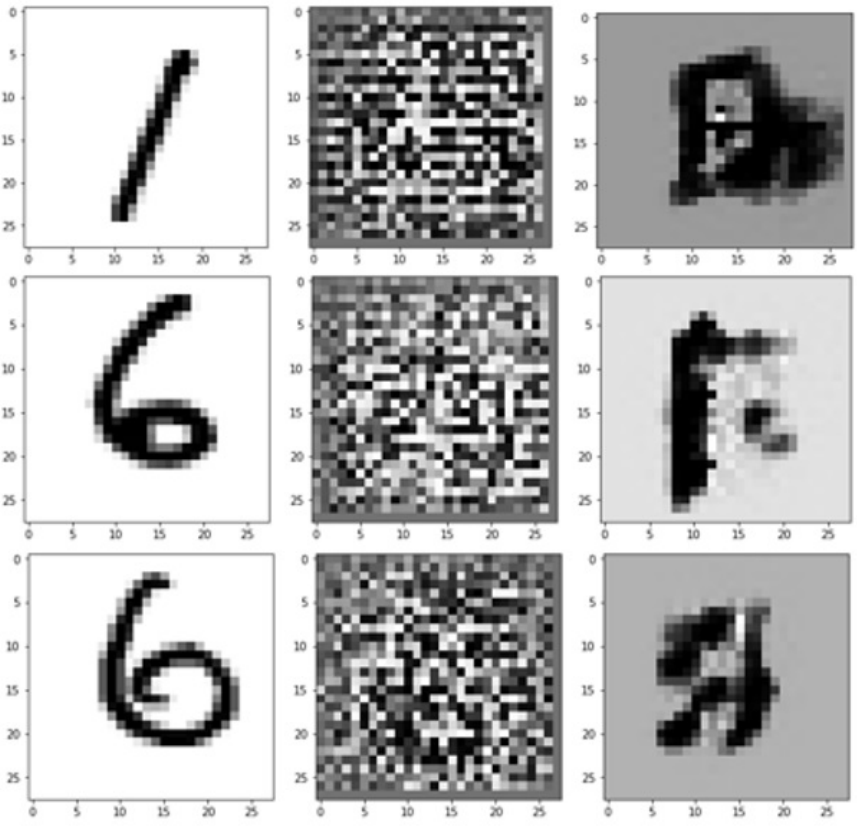

Figure 5.11: Output after the first initial training stage

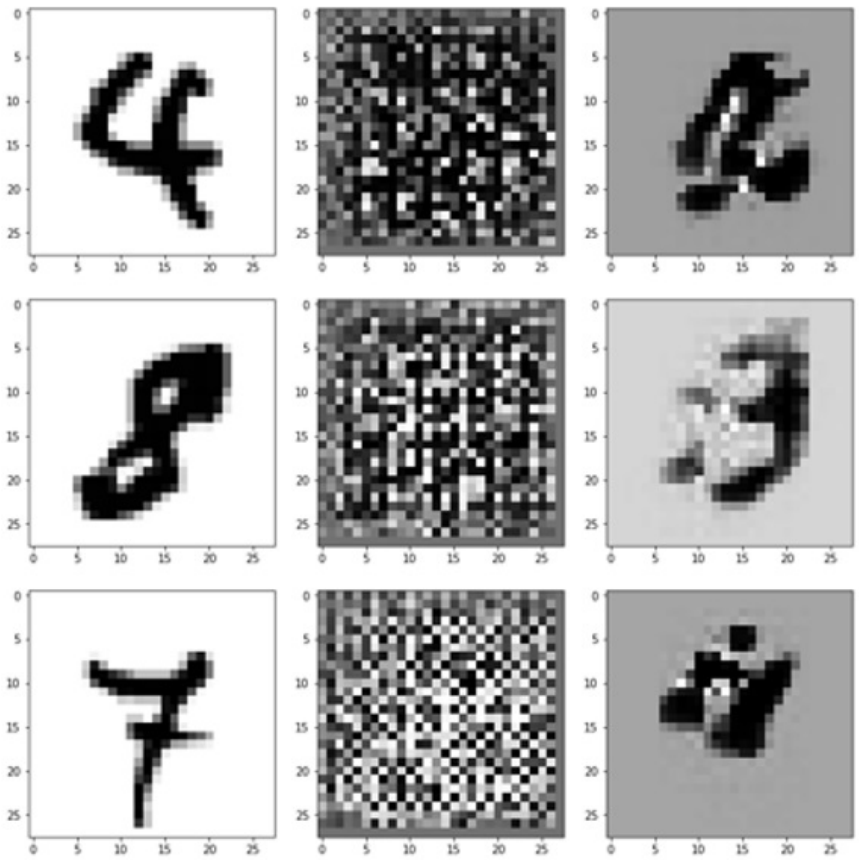

Figure 5.12: Output in an advanced stage of training loops

5.6.1.2 Experimental setup

In this work, we used a Keras GAN. Keras implementations of Generative Adversarial Networks (GANs) suggested in many research papers and it can be found in "https://github.com /eriklindernoren/Keras-GAN". Also, we used for training task the TensorFlow backend. • Some analysis: As was mentioned, the training tips for the combined model for the discriminator and the generator as follows: (False = 0, True = 1). a- Train the discriminator:

$$D_{Loss} = \frac{LossReal + LossFack}{2} \tag{5.14}$$

b- Train the generator (to have the discriminator label samples as valid):

G Loss = combined train (noise, valid)

Noise = random batch on image

Valid = adversarial ground truth

In this experiment, we will use two different pieces of training: a- Using pretrained weights file (hd5) for the previous dataset. b- Pretrain the dataset and use its own weights file (hd5). c- Compare between the two cases. Our first task was to prepare the dataset. There are two classifications in the training dataset: forged and pristine. The paired images will be used to illustrate how the forgery operation proceeds (here, copy-move forgery). Prior to starting the training, pixels from multiple images are loaded into a directory using the NumPy array distribution. We begin with the pristine category and then perform tests with both forged and pristine images. To obtain a directory image file list, we import the glob, use file-list for constructing two diminutions matrix, and then convert the file-list to a NumPy array. Thus, if we combine the total images as one NumPy array, we get:

$$x = np.array([np.array(Image.open(f - name))for f - nameinfile - list]) \quad (5.15)$$

This array divided later to three partitions for training, verification, and testing.

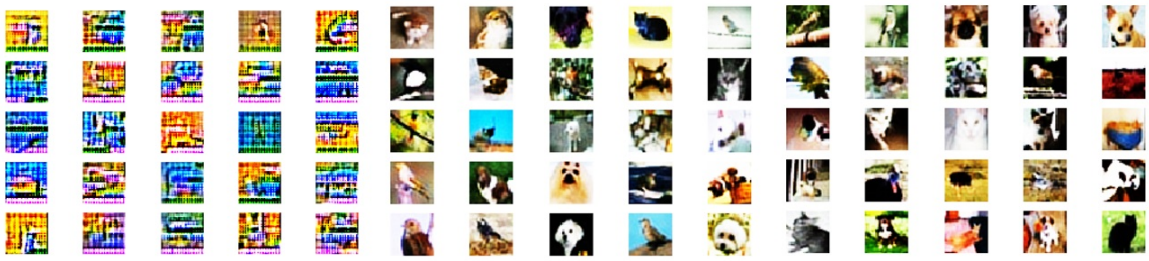

Figure 5.13: From left to right, some samples from random results show the process of training improving transition result of the model using local dataset

.

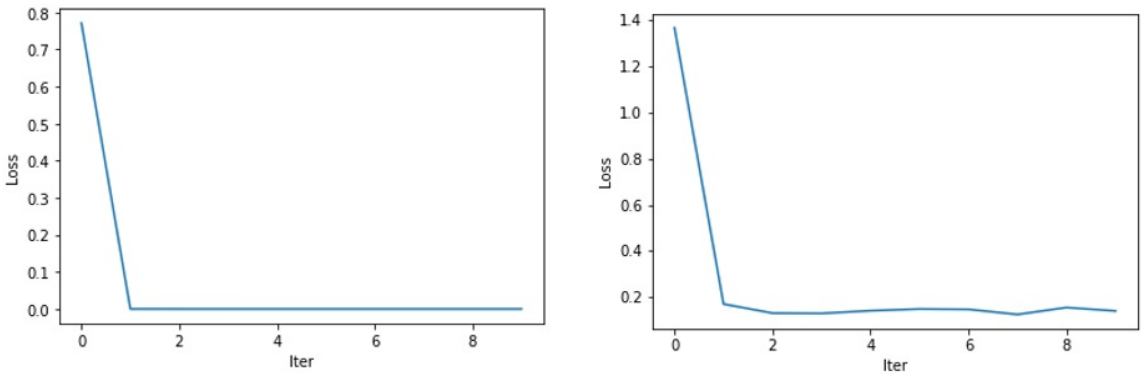

Figure 5.14: Loss functions (D loss, G loss) of the training model using a custom local dataset used pretrained discriminator and generator respectively

.

.

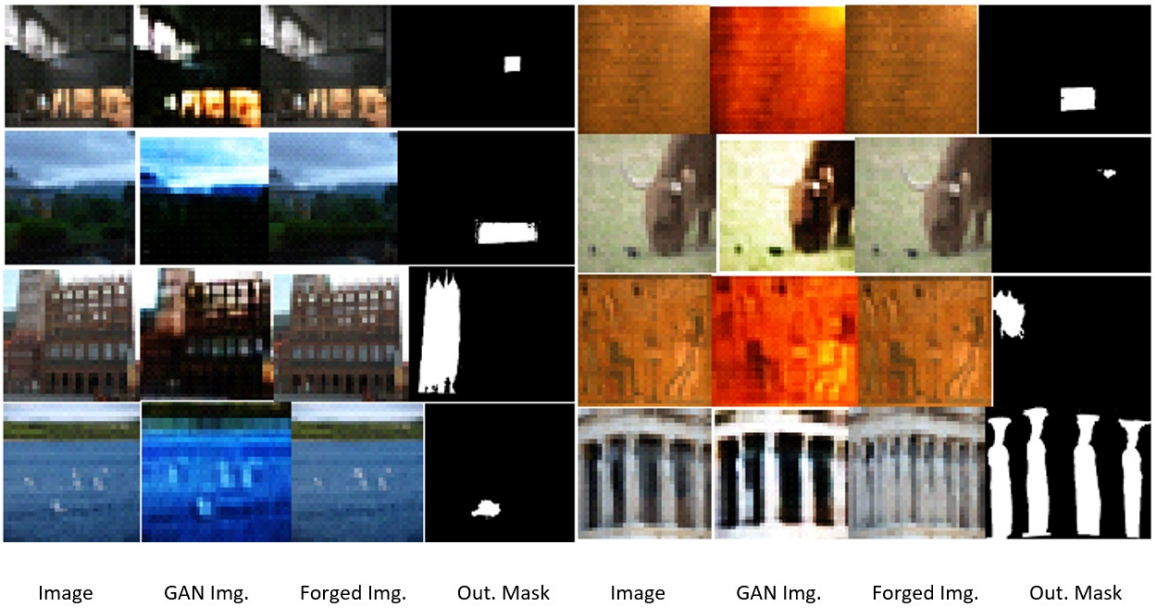

Figure 5.15: Detect the copy-move forgery using GAN model

5.6.2 Training CNN models in similarity detection

In this study, a new type of deep neural network architecture was introduced for detecting and localizing copy-move forgery. The model, which uses a supervised end-to-end trainable network, was able with high accuracy to detect the copy-move forgery. Specifically, it was able to pinpoint areas in the image that were sourced from the same original area.

5.6.2.1 How the model works

The model functions as follows. First, an image I is input, from which features are extracted by employing the CNN feature extractor. Next, based on the extractions, feature similarity is calculated using the self-correlation module, and statistics are gathered using percentile pooling. These then upsample feature maps against the original image size by applying a mask decoder, after which a binary classifier creates a copy-move mask. The present research gained inspiration from a number of published

works, especially [43][63][39].

To sum up, the copy-move forgery detection baseline was first developed by making feature maps from input image extracts, followed by the construction of relevant feature statistics on the basis of percentage pooling process from upsampled feature maps. The feature classifier was then applied as a means to doctor and assign similar regions as copy-move forgery areas. The subsequent tests we performed relied on a pre-trained weight file for the initial model branch; later, however, we built another weight file (HD5 format), intending to compile into a single entity the large amount of images. This made our training more accurate and readied the stage for the subsequent portions of the model construction. Numerous image samples were used in the training process. A single-class classifier tested every image to determine whether it was a copy-move forgery or pristine. Figures 5.16 and 5.17 depict images that are forged as well as a detected soft mask and genuine mask. The test applied a set of forged images to carry out the primary task of the network branch, using recall, precision and F1 scores for reporting CMFD performance. We considered the classified pixels from both the target and the source as being forged in order to compare our proposed model to other CMFD approaches that predict only binary masks.

.

.

5.6.2.2 Some result using different datasets

Numerical results in table 5.2 are shown discernibility summery of different datasets form state-of-the-art compared to the proposed model, while the other hand figure 5.18 gives a visual illustration of similar comparison.

.

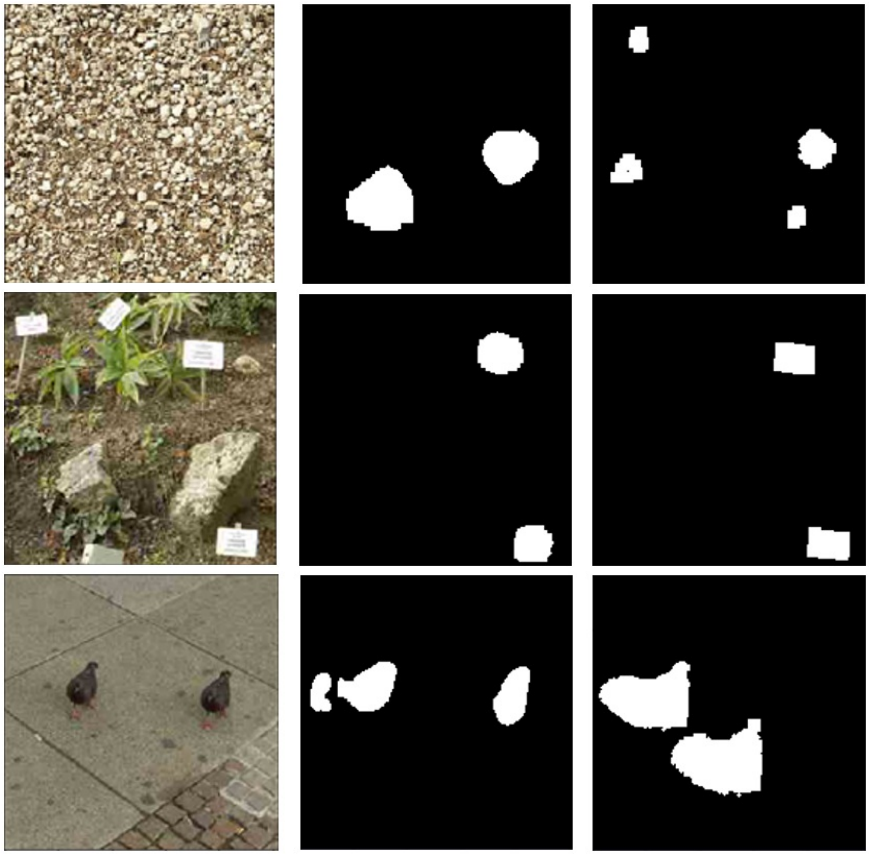

Figure 5.16: Shows random results with F1 score for T greater than 0.25

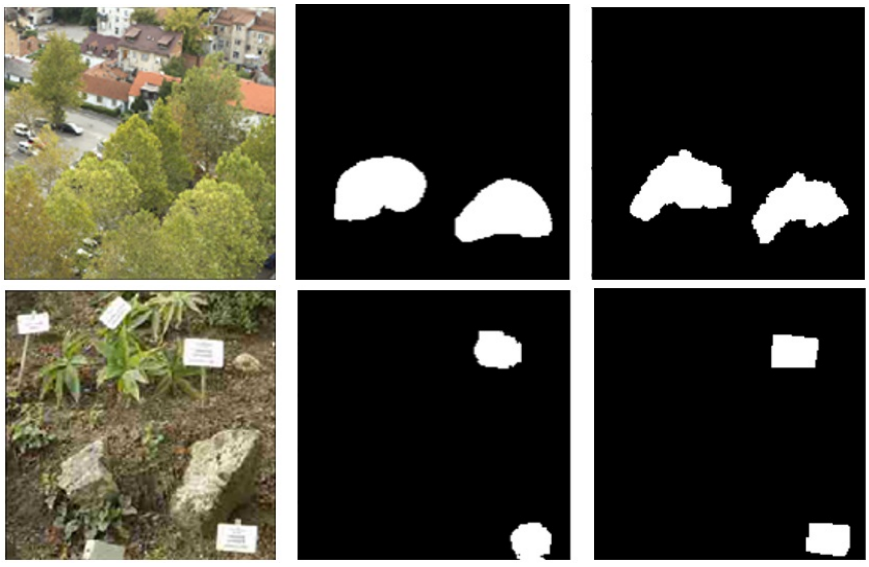

Figure 5.17: Shows random results with F1 score for T greater than 0.75

Table 5.2: Shows the skeleton metrics for the precision recall F1 scores

Algorithm	[37]	[94]	[23]	[106]	[64]	Proposed
F1	0.4926	0.5439	0.5943	0.6055	0.6318	0.8835
Precision	0.5734	0.5390	0.5440	0.5662	0.5927	0.6963
Recall	0.4939	0.8327	0.8020	0.8040	0.8220	0.8042

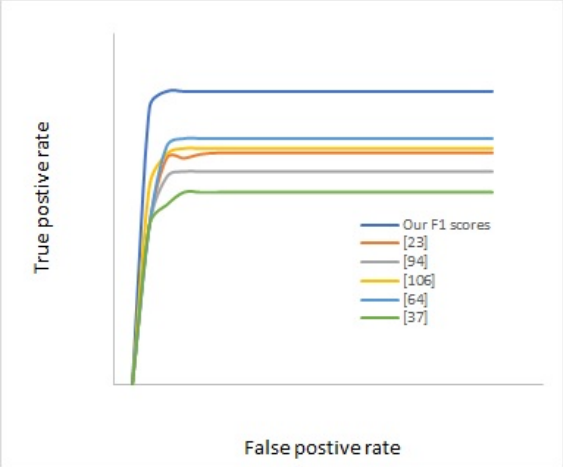

Figure 5.18: Shows the ROC for F-scores comparison

5.6.3 Training the CMFD classification model for localization

A single support vector machine (SVM) classifier was employed in our model for all three branches for detection and classification. The outcomes indicate precision, accuracy, and good recall and F-measure. The primary aim of the model's third branch was verifying and then localizing any detected copy-move forgery. Note that the first network uses GAN for detecting and manipulating areas of the frame of the input image, whereas the second branch detects similar areas of the input image as being indicative of a copy-move forgery incidence. The third branch merges the outputs from the two other branches, as follows:

5.6.3.1 Verifying CMFD

Image forgery is detected by the GAN network through determining one or more interrupted areas, as detailed in a previous section (6.1). Then, as mentioned in section (6.2), the CNN network matches similar batches in the image. This step trains the merging network to detect any forgery through comparing and testing the outputs.

5.6.3.2 Localizing CMF

Following the verification of CMF detection, the model constructs a mask which pinpoints the CMFD location within the forged image frame.

Figure 5.19: a- Masking the forged area (GAN), b- Masking the similar areas in the forged image frame (CNN), c- Forgery location

.

Evaluating forgery localization entails a similar process in images where forgery has already been discerned. In these cases, the mask estimates $M^{'}$ indicate the threshold but subsequently are compared (pixel-wise) with related ground truth masks M. Note that the ROC curves here Figure 5.20 indicate localization for various threshold levels. By applying the Binary Cross Entropy BCE loss, we can see that the PR curve obtains 0.6963 as a mean precision score and 0.8042 as a mean recall score, confirming the

strength of the localization results.

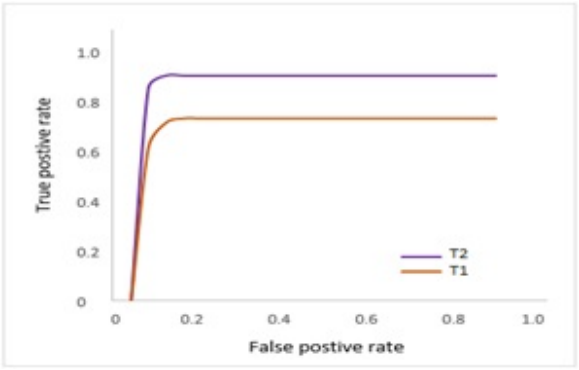

Figure 5.20: Fig 6. Shows the ROC for F-scores

.

The precision recall area under the curve (AUC) Figure's 5.21 illustrates excellent detection accuracy. Here, the target is to have the model curve in the upper point in the right corner, which presents an ideal model with 100% true positive and zero false positive rates, regardless of recall. The area under the curve shows that perfect detection of the forgery using this model is fairly likely.

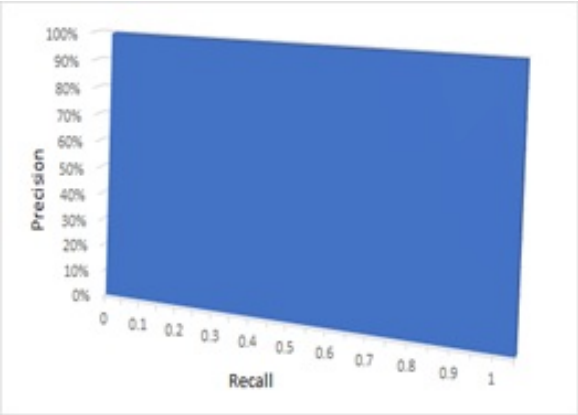

Figure 5.21: Illustrates that the area under the curve (AUC)

.

5.6.3.3 Determining the CMF area vs. source area

Because the variations between the source (original) and the copy-move batch can be very small, the unaided human eye is usually unable to detect the forgery. Therefore, auto-detection with a deep learning method is necessary to determine a forged image. This is especially important for discerning an original source image from a forged one. The figure illustrates the source and CMF areas in various colors, making it easier for people to see the difference between the images.

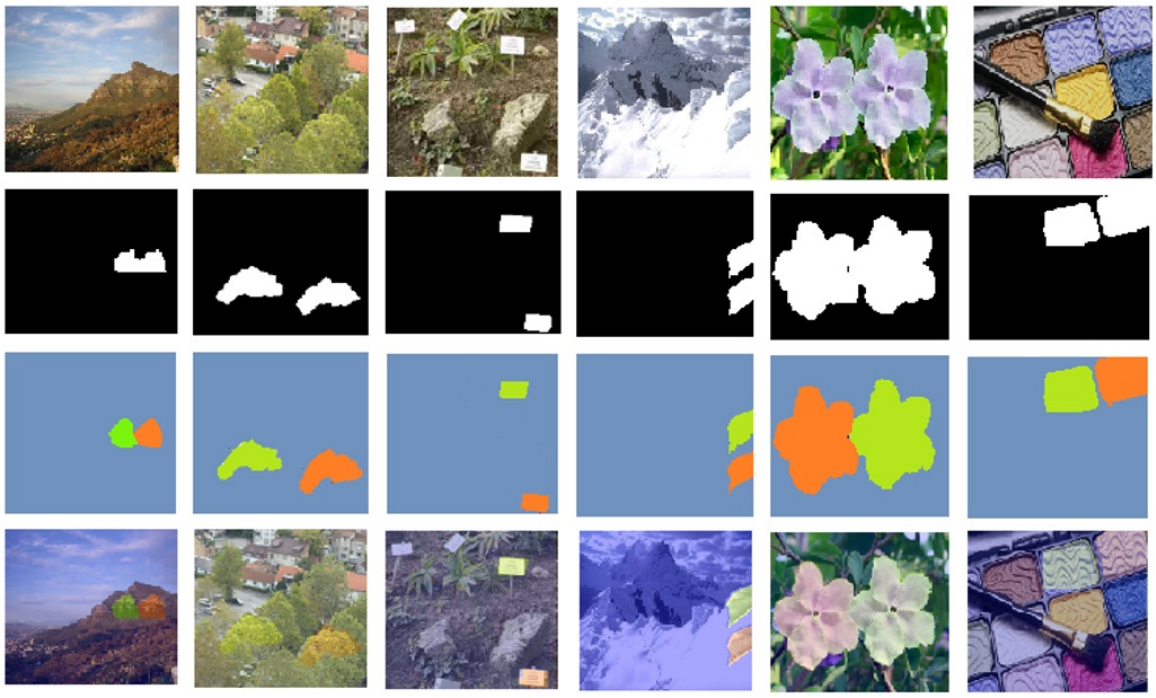

Figure 5.22: Shows the output of the model

.

5.7 Conclusions

The present work proposed an image analysis method that employs GAN for localization and splicing detection of images. The novel method adopts a data-driven

strategy that enables the algorithm to constantly update its learning via training data on how to discern pristine areas from forged ones. The test results clearly indicate the significantly high accuracy of the proposed method in applying the dataset in the discernment of localization and tampering detection. Also noteworthy is the proposed technique's ability to project its findings to differently sized forgeries than the ones used in training. The encouraging results from these tests encourage an extension of the research to additional related inquiries, such as testing the method with other kinds of forgeries and datasets. The rationale for the proposed CMF solution in the present work is testing the potential for training auto-encoders to acquire a representation of image patches originating in pristine images. Such an auto-encoder could also be employed in feature extraction of image patches. In the tests, a single-class SVM detects if feature vectors are derived from pristine or forged images. Generative adversarial networks are used for training the auto-encoder in detecting forgery, with the entire system being trained solely with pristine data. The system thus has no prior knowledge of forgeries. Even so, the test results indicate a high level of accuracy for localization as well as detections. In fact, the proposed model succeeded in its assigned CDFD task, giving performance results in the 93% to 97% range. Future related work could consider testing and enhancing system robustness by introducing a variety of different forgery types and increasing the number of used images for GAN training by including more datasets, such as ImageNet. The sensitivity of the presented methodology to the setup of its parameters should be investigated.

5.8 References

[1] A. Rocha, W. Scheirer, T. Boult, and S. Goldenstein, "Vision of the unseen: Current trends and challenges in digital image and video forensics," ACM Computing Surveys, vol. 43, pp. 1–42, October 2011.

[2] Piva, "An overview on image forensics," ISRN Signal Processing, vol. 2013, pp. 22, November 2013.

[3] C. Stamm, Min Wu, and K. J. R. Liu, "Information forensics: An overview of the first decade," IEEE Access, vol. 1, pp. 167–200, May 2013.

[4] Fridrich, A.J., Soukal, B.D., Luk´aˇs, A.J, "Detection of copy-move forgery in digital images," in Proceedings of Digital Forinsic Research Workshop. Citeseer, 2003.

[5] Ke, Y.,Sukthankar, R., Huston, L., "An efficient parts-based near-duplicate and sub-image retrieval system," in Proceeding of the 12th annual ACM international conference on Multimidia," pp. 869-876. 2004.

[6] Yue Wu; W. Abd-Almageed; and P. Natarajan, "BusterNet: Detection Copy-Move Image Forgery with Source / Target Localization," ECCV, 2018.

[7] S. Fan, T.-T. Ng, B. Koenig, J. Herberg, M. Jiang, Z. Shen, and Q. Zhao, "Image visual realism: From human perception to machine computation," IEEE Transactions on Pattern Analysis and Machine Intelligence, in press, 2017.

[8] S. Lyu and H. Farid, "How realistic is hotorealistic?" IEEE Transactions on Signal Processing, vol. 53, no. 2, pp. 845 – 850, 2005.

[9] R. Wu, X. Li, and B. Yang, "Identifying computer generated graphics via histogram features," in IEEE ICIP, pp. 1933–1936. 2011.

[10] S. Dehnie, H. Sencar, and N. Memon, "Digital image forensics for identifying computer generated and digital camera images," in IEEE ICIP, 2006, pp. 2313–2316.

[11] A. Dirik, S. Bayram, H. Sencar, and N. Memon, "New features to identify computer generated images," in IEEE ICIP, , pp. IV–433–IV–436. Oct 2006.

[12] A. Gallagher and T. Chen, “Image authentication by detecting traces of demosaicing,” in IEEE CVPR Workshops, June, pp. 1–8. 2008.

[13] J.-F. Lalonde and A. Efros, “Using color compatibility for assessing image realism,” in IEEE ICCV, pp. 1–8. Oct 2007.

[14] R. Zhang, R.-D. Wang, and T.-T. Ng, “Distinguishing photographic images and photorealistic computer graphics using visual vocabulary on local image edges,” in International Workshop on Digital Forensics and Watermarking, pp. 292–305. Oct 2011.

[15] N. Rahmouni, V. Nozick, J. Yamagishi, and I. Echizeny, “Distinguishing computer graphics from natural images using convolution neural networks,” in IEEE WIFS, pp. 1–6. 2017.

[16] E. de Rezende, G. Ruppert, and T. Carvalho, “Detecting computer generated images with deep convolutional neural networks,” in SIBGRAPI Conference on Graphics, Patterns and Images, pp. 71–78. 2017.

[17] O. Holmes, M. Banks, and H. Farid, “Assessing and improving the identification of computer generated portraits,” ACM Transactions on Applied Perception, vol. 13, pp. 1–12, 2016.

[18] J. Thies, M. Zollh¨ofer, M. Stamminger, C. Theobalt, and M. Nießner, “Face2Face: Real-Time Face Capture and Reenactment of RGB Videos,” in IEEE CVPR, pp. 2387-2395, 2016.

[19] N. Haouchine, F. Roy, H. Courtecuisse, M. Nießner, and S. Cotin, “Calipso: Physics-based image and video editing through cad model proxies,” arXiv preprint arXiv:1708.03748, 2017.

[20] Bayram, S., Sencar, H.T., Memon, N. “An efficient and robust method for detecting copy-move forgery.” In: Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on. pp. 1053–1056. IEEE, 2009.

[22] Huang, D.Y., Huang, C.N., Hu, W.C., Chou, C.H. “Robustness of copy-move forgery detection under high jpeg compression artifacts.” Multimedia Tools and Applications 76(1), 1509–1530, 2017.

[23] Mahdian, B., Saic, S. “Detection of copy–move forgery using a method based on blur moment invariants.” Forensic science international 171(2), 180–189, 2007.

[24] Mahmood, T., Nawazm, T., Irtaza, A., Ashraf, R., Shah, M., Mahmood, M.T. “Copy-move forgery detection technique for forensic analysis in digital images.” Mathematical Problems in Engineering 2016.

[25] Ryu, S.J., Lee, M.J., Lee, H.K. “Detection of copy-rotate-move forgery using Zernike moments.” In: Information hiding. vol. 6387, pp. 51–65. Springer 2010.

[26] Ardizzone, E., Bruno, A., Mazzola, G. “Copy–move forgery detection by matching triangles of keypoints.” IEEE Transactions on Information Forensics and Security 10(10), 2084–2094, 2015.

[27] Manu, V., Mehtre, B.M. “Detection of copy-move forgery in images using segmentation and surf.” In: Advances in Signal Processing and Intelligent Recognition Systems, pp. 645–654. Springer 2016.

[28] Shivakumar, B., Baboo, S.: Detection of region duplication forgery in digital images using surf. International Journal of computer science Issues 8(4), 199–205, 2011.

[29] Silva, E., Carvalho, T., Ferreira, A., Rocha, A. “Going deeper into copy-move forgery detection: Exploring image telltales via multi-scale analysis and voting processes.” Journal of Visual Communication and Image Representation 29, 16–32, 2015.

[30] Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A., Serra, G. “A sift-based forensic method for copy–move attack detection and transformation recovery.” IEEE Transactions on Information Forensics and Security 6(3), pp. 1099–1110, 2011.

[31] Yang, B., Sun, X., Guo, H., Xia, Z., Chen, X. “A copy-move forgery detection method based on cmfd-sift.” Multimedia Tools and Applications pp. 1–19, 2017.

[33] Pun, C.M., Yuan, X.C., Bi, X.L. “Image forgery detection using adaptive over segmentation and feature point matching.” IEEE Transactions on Information Forensics and Security 10(8), 1705–1716, 2015.

[34] Asghar, K., Habib, Z., Hussain, M. “Copy-move and splicing image forgery detection and localization techniques: a review.” Australian Journal of Forensic Sciences 49(3), 281–307, 2017.

[35] Birajdar, G.K., Mankar, V.H. “Digital image forgery detection using passive techniques: A survey.” Digital Investigation 10(3), 226–245, 2013.

[36] Soni, B., Das, P., Thounaojam, D. “Cmfd: A detailed review of block based and key feature-based techniques in image copy-move forgery detection.” IET Image Processing, 2017.

[37] Warif, N.B.A.,Wahab, A.W.A., Idris, M.Y.I., Ramli, R., Salleh, R., Shamshirband, S., Choo, K.K.R. “Copy-move forgery detection: Survey, challenges and future directions.” Journal of Network and Computer Applications 75, 259–278, 2016.

[38] Bunk, J., Bappy, J.H., Mohammed, T.M., Nataraj, L., Flenner, A., Manjunath, B., Chandrasekaran, S., Roy-Chowdhury, A.K., Peterson, L. “Detection and localization of image forgeries using resampling features and deep learning.” In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on. pp. 1881–1889. IEEE, 2017.

[39] Wu, Y., Abd-Almageed,W., Natarajan, P. “Deep matching and validation network: An end-to-end solution to constrained image splicing localization and detection.” In: Proceedings of the 2017 ACM on Multimedia Conference. pp. 1480–1502. MM ’17, 2017.

[40] Zhou, P., Han, X., Morariu, V.I., Davis, L.S. “Two-stream neural networks for tampered face detection.” In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on. pp. 1831–1839. IEEE, 2017.

[41] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde- Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in neural information processing systems, 2014, pp. 2672–2680.

[42] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in IEEE ICCV, 2017.

[43] O. Carey, 'Generative Adversarial Networks GANs', 2018. [Online]. http://towardsdatascience.com/generative-adversarial-networks-gans-a-beginners-guide-5b38eceece24. [Accessed: 19- Feb- 2019].

[44] Fridrich, A.J., Soukal, B.D., Luk´a?s, A.J, "Detection of copy-move forgery in digital images," in Proceedings of Digital Forinsic Research Workshop. Citeseer, 2003.

[45] Ke, Y.,Sukthankar, R., Huston, L., "An efficient parts-based near-duplicate and sub-image retrieval system," in Proceeding of the 12th annual ACM international conference on Multimidia," pp. 869-876. 2004.

[46] Yue Wu; W. Abd-Almageed; and P. Natarajan, "BusterNet: Detection Copy-Move Image Forgery with Source / Target Localization," ECCV, 2018.

[47] J. Thies, M. Zollh¨ofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2Face: Real-Time Face Capture and Reenactment of RGB Videos," in IEEE CVPR, pp. 2387-2395, 2016.

[48] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-toimage translation with conditional adversarial networks," in IEEE CVPR, 2017.

[49] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in IEEE ICCV, 2017.

[50] J. Fridrich, D. Soukal, and J. Luka´?s, "Detection of copy-move forgery in digital images," in Proceedings of Digital Forensic Research Workshop, Cleveland, Ohio, USA, August 2003.

[51] S. Bayram, H. T. Sencar, and N. Memon, "An efficient and robust method for

detecting copy-move forgery," in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '09), pp. 1053–1056, April 2009.

[52] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," Tech. Rep. TR2004-515, Dartmouth College, Hanover, NH, USA, 2004.

[53] Y. Huang, W. Lu, W. Sun, and D. Long, "Improved DCT-based detection of copy-move forgery in images," Forensic Science International, vol. 206, no. 1–3, pp. 178–184, 2011.

[54] T. Mahmood, T. Nawazm, R. M. Shah, and M. Mahmood, "Copy-Move Forgery Detection Technique for Forensic Analysis in Digital Images," Hindawi Publishing Corporation Mathematical Problems in Engineering, pp. 1-13, 2016.

[55] D. Cozzolino, D. Gragnaniello, L. Verdoliva, "Image forgery detection based on the fusion of machine learning and block-matching methods." Computer Science - Computer Vision and Pattern Recognition, arXiv: 1311.6934C. pp. 1-4, 2013.

[56] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5967–5976, July 2017. Honolulu, HI.

[57] S., Kalyan Y., D. Guera, P. B., F. Maggie, S. Tubaro, E. J. Delp, "Satellite Image Forgery Detection and Localization Using Gan and One Class Classification," in Media Watermarking, security, and Forensics. ArXiv: 1802.04881v1, 2018.

[58] Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Advances in Neural Information Processing Systems. pp. 3859–3869, 2017.

[59] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. "Gradient-based Learning Applied to Document Recognition." Proceedings of the IEEE 86(11), 2278–2324, 1998.

[60] Im, D.J., Kim, C.D., Jiang, H., Memisevic, R.: Generative adversarial metric, 2016.

[61] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Nets. In: Advances in neural information processing systems. pp. 2672–2680, 2014.

[62] Hinton, G.E., Krizhevsky, A., Wang, S.D.: Transforming auto-encoders. In: International Conference on Artificial Neural Networks. pp. 44–51. Springer, 2011.

[63] Dwibedi, D.,Misra, I., Hebert, M. "Cut, paste and learn: Surprisingly easy synthesis for instance detection." In: The IEEE International Conference on Computer Vision (ICCV), 2017.

[64] Silva, E., Carvalho, T., Ferreira, A., Rocha, A.," Going deeper into copy-move forgery detection: Exploring image telltales via multi-scale analysis and voting processes," Journal of Visual Communication and Image Representation 29, pp. 16–32, 2015.

[65] Li J, Li X, Yang B, Sun X. "Segmentation-based image copy-move forgery detection scheme". IEEE Trans Inform Forens Secur 10(3). pp. 507–518. 2015.

[66] Yaqi L., Q. Guan, X. Zhao, "Copy-move Forgery Detection based on Convoluational Kernel Network," arXiv:1707.01221[cs.CV], pp. 1-26, 2017.

[67] Liu, Y., Guan, Q., Zhao, X.: Copy-move forgery detection based on convolutional kernel network. Multimedia Tools and Applications pp. 1–25, 2017.

[68] O. Ronneberger, P. Fischer, and T. Brox. "U-Net. Convolutional networks for biomedical image segmentation." Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 234– 241, October 2015. Munich, Germany.

[69] Krizhevsky, A. "Learning multiple layers of features from tiny images", 2009.

Chapter 6

Image Forgery Detection Based on Deep Transfer Learning

Preface.

A version of this chapter has been published in the Journal of EJECE, European Journal of Electrical and Computer Engineering 2019; 5: Isue 3, 1-8. I am the primary author. Along with Co-authors, Tariq Iqbal and M. Shehata, I conceptualized the idea and developed the conceptual algorithm. I have prepared the first draft of the manuscript and subsequently revised the manuscript, based on the feedback from Co-authors and also peer review process. As Co-authors, Tariq Iqbal and M. Shehata assisted in developing the concept and testing the algorithm, reviewed and corrected the model and results. Also, the Co-author T. Iqbal critically reviewed the content and revising the manuscript. Contribution was made through the team support to developing and testing the algorithm as well as and reviewing the manuscript.

Abstract – The recent digital revolution has sparked a growing interest in applying convolutional neural networks (CNNs) and deep learning to the field of image

forensics. The proposed methods aim to train algorithms for solving a range of predetermined tasks. However, training a model that has been randomly initialized requires extensive time for computation as well as an enormous pool of training data to draw from. Moreover, such a model needs to be developed and redeveloped from the ground up if there are any alterations to the feature-space distribution. In addressing these problems, the present paper proposes a novel approach to training image forgery detection models. The method applies prior knowledge that has been transferred to the new model from previous steganalysis models. Additionally, because CNN models generally perform badly when transferred to other databases, transfer learning accomplished through knowledge transfer allows the model to be easily trained for other databases. The various models are then evaluated using image forgery techniques such as shearing, rotating, and scaling images. The experimental results, which show an image manipulation detection has validation accuracy of over 94.89%, indicate that the proposed transfer learning approach successfully accelerates CNN model convergence but does not improve image quality.

Keywords: Forgery detection; Deep learning; Transfer learning; Neural network.

6.1 Introduction

Human beings are generally hard-wired to apply or transfer knowledge that they have learned in one skill to other related skills. In this way, acquired knowledge can help solve problems in less time through the cross-utilization of knowledge which occurs during these transfers [1]. Considering this built-in problem-solving framework, let us look at two critical problems currently restricting progress in the field of image

forensics. These problems are: (1) developing a sufficiently large and diverse body of annotated images for use as training models; and (2) including in the models non-image data in order to permit and enable broader application of the model(s). This research will address both of these challenges. Regarding the creation of a large body of images, this can be an extremely costly undertaking, as the task requires humans rather than machines to do the image classification. Even so, the workers tasked with this job could be severely challenged by the image complexity and vast array of different classes for categorization. Regarding issues around incorporating non-image data in the models, researchers and other workers in the field still experience problems when trying to extract image data from images that also incorporate non-image data, such as text. However, finding a way to include non-image data in transfer learning is crucial if the models are to be applied to areas such as the medical field or insurance industry. In overcoming these challenges, the present work will employ text metadata to deal with noisy classifications in images. Within these datasets are forged images which will comprise the metadata. So, instead of attempting image classification or categorization, it will be assumed that the images share certain metadata features along with some aspects of feature representations. Two distinct image categories – pristine images and forged images – will be presented, with the metadata being sourced from the Internet. Although this will enable us to substantially expand our available training data set, it will also likely mean that we will be including some forged labels as a trade-off. We will use cosine similarity to gauge similarities within the metadata, even though this will result in some compromise in feature representation quality and similarity complexity that the system learns. This is another trade-off that is considered negligible. However, there is a relatively significant problem with this strategy – namely, the similarities that exist between the pristine image and forged image metadata. These similarities exist because the Internet-sourced data

were not developed as image classifiers but instead were specifically created as fakes of original images. We note that, though beyond the scope of the present work, numerous applications have shared and related images as well as repeatable features that have similar or even the same colors (e.g., claims in the insurance field that feature damaged vehicle images). These existing models can already incorporate both images and text as data, but they are also prohibitively expensive to build and maintain. This is because large training datasets must be employed to compensate for noisy labels. So, while acknowledging the existence of these models, our work aims to utilize datasets in combination [2], as this approach requires smaller datasets and shows promise of high image quality. Moreover, our study relies on the concept of transfer learning as a means to reduce both the amount of resources required and the time it takes to train the networks. In our experiments, we initialize the networks by applying basic weights learned in earlier large-scale training, such as CaffeNet and VGG16 [1][3]. We can see by the accuracy in the baseline classification task (i.e., cat versus dog images) that the convolutional neural network (CNN) weights are able to give high-quality feature representation on diverse image datasets. So, our baseline here will be to develop a "cat versus dog" image classifier through fine-tuning CNN weights as a means to significantly shorten the computational time required to train the network. We will run the network using an extensive sampling of image pairings, with the network learning similarity between the provided images, both forged and pristine. As well, we will fine-tune the CNN weights in order to achieve feature representations with minor differences that are not so different that they require a full re-training. Furthermore, because CNN middle layers and baseline architecture are alike, we aim to demonstrate the practicality and effectiveness of re-training the baseline by applying the original network weights instead of new CNN weights, with the aim of enhancing forgery detection abilities with more accurate image classification.

This chapter covers the strategy used to develop a new classification model meant to detect the forged images in digital dataset using pretrained model. This work is organized as follows. Following the introduction, we will provide an overview and revision of the related works. In the subsequent sections, we will dive deeper into the topics mentioned in the overview training, performance testing, and also validation of the used methods.

6.2 Related Works

The present paper presents a full review of the background, foundational ideas, and current applications for transfer learning, including both historical and recent examples. It is generally well-known in the industry that transfer learning evolved from machine learning as well as statistical modeling. With this in mind, we look at some research conducted by Han et al. [4] and Doersch et al. [5]. Additionally, we review MatchNet and the study of D. Itera [6], noting that their two-tower architecture shares some similarities with that used in the present work. Specifically, the towers share weights that are first concatenated; the resulting feature representations for patch pairs are then relayed through the metric network (fully connected) and SoftMax loss function. In the metric network, similarities between the two features representations are measured, after which the ground truth similarity loss of the patch pair is formulated. The outcome is equivalent to the present work's formulation for similarities in the two feature representations; however, the ground truth towers rely on the text metadata accompanying the images instead of relying on a computed function overlaying the images. In [5], Doersch et al. study the behavior of learning features in unsupervised datasets. They divide each image to create a "patch" and then train networks to figure out the right (i.e., original) orientation among the patched pieces.

The aim here is to see whether or not a network is able to learn objects occurring within images; if a network can do this, it indicates an ability to learn underlying feature orientation for images. Therefore, by applying these learned representations, features learned from this network could potentially be reused in unsupervised object detection for other datasets. It is important to note that transfer learning, as mentioned, is a concept which evolved from machine learning and statistical modeling. More recently, transfer learning has been investigated for its application in deep learning. However, earlier approaches that were previously employed to construct and train machine learning models differ significantly from methodologies that adhere to transfer learning strategies. The present work aims to locate similarities in "similar" images' underlying map features by applying the latest deep learning techniques.

6.3 Transfer learning strategies

The type of transfer learning strategy that is most suitable for a given problem is determined by several factors, including data availability, the task to be performed, and the domain. In general, transfer learning methods are classified according to the kind of conventional ML algorithms that are used. The three main categories explored in this work are unsupervised transfer learning, transductive transfer learning, and inductive transfer learning, as explained below. The first category, unsupervised transfer learning, deals with unsupervised tasks located in target domains. Although both the target and the source domains could potentially be similar, their tasks are quite dissimilar. Later in this study, we will work on labeled data which has been made unavailable at both domains. The second category to be explored here is transductive transfer learning, which is employed when there are similarities between target and source tasks but dissimilar corresponding domains. In this scenario, there is no

labeled data in the target domain, whereas the source domain contains sample labeled data. Transductive transfer learning can also include subcategories regarding settings (e.g., marginal probabilities or different feature spaces). In inductive transfer learning, which will be the third category studied, both the target and source domains are the same, even though their respective tasks differ. In the inductive transfer learning setting, algorithms employ the source domain's inductive biases as a means to make an improvement to the target tasks. This setting can be divided into the subcategories of self-taught learning and multitask learning, according to whether there is labeled data in the source domain or not. The source task's inductive biases help to carry out the target task. As shown in the figure 6.1 below, this is accomplished by making adjustments to the target task's inductive biases through the restriction of model space, adjusting the search process using the knowledge acquired from the source task, or limiting the hypothesis space.

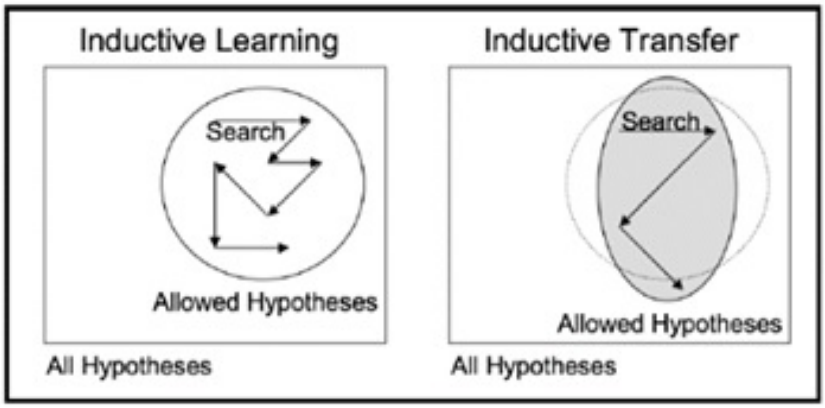

Figure 6.1: Inductive transfer techniques of a target [1]

.

6.4 Transfer learning approaches

We will demonstrate in our work that a few of the approaches are able to be used in the aforementioned strategies, as follows: (1) Instance transfer: This involves reusing (or re-purposing) knowledge obtained from a source domain in a target task. However, in many instances, we cannot directly reuse source domain data, but we might be able to reuse this data in tandem with target data. For inductive transfer cases, we can apply modifications like AdaBoost (Dai et al.) to assist with improvements to target tasks when training from source domains. (2) Relational-knowledge transfer: This approach deals with non-IID data (e.g., data which are not identically distributed or independent). These type of data have data points that are related to other (sometimes similar) data points. A good example of relational-knowledge transfer in the current application is social network data. (3) Parameter transfer: In the parameter transfer approach, it is assumed that models used for related tasks involve a few or more shared parameters and/or hyperparameters. (4) Feature-representation transfer: The purpose of this technique is to mitigate error rates and domain divergence through the identifications of positive (good) feature representations. Such feature representations can then be used for target domains from source domains. Instances, where feature-representation transfer is used, include supervised/unsupervised methods, if there is sufficient labeled data available.

6.5 Similarity detection

In the similarity detection process, the similarity is measured by taking any related metadata text applied to an image pair, from which the score of 0 to 1 indicates the similarity of the images, according to the detected features in the objects measured. Although features which are similar will probably produce similar results, repeated

features (e.g., neighborhood or background features) do not always give similar results. Furthermore, this function should be sufficiently robust to compare large and small areas, as patches of various sizes need to be comparable. However, this operation can be very costly, given how many pairs are able to be generated in a dataset and considering that the operation typically runs several times. At the outset, the present work employs cosine similarity, which offers users a good similarity measure for comparing sets of varying sizes. However, a trade-off between results accuracy and computational complexity is expected and understood between cosine similarity and other more complicated and time-consuming approaches. Using cosine similarity, two feature "bags" will be developed with feature sets related to the images as well as to the counts for every feature. This model is expected to be biased toward images that are dissimilar, given that dissimilarity has been shown to be more common. As well, we will construct a preprocessing engine which is able to compute image similarity and from that to develop balanced pairs out of the similar/dissimilar images. These will then be labeled accordingly as contradictory categories. Note that this component also includes similarities between different texts and is able to be expanded as needed. If we obtain positive results from the simplified similarity function, we would consider carrying out a future study focusing on the kind of trade-offs incurred between using highly complex similarity models and simpler ones, based on effectiveness and runtime. This, however, is outside the scope of the present work, whose main aim is to test similar and non-similar images, not performance grades of similarity.

6.6 The proposal transfer learning approach

The present work mainly investigates a range of deep learning models. Although the various techniques mentioned above are applicable in different instances of machine

learning, is transfer learning also applicable to deep learning? Deep learning models can best be described as being inductive learning approaches. As touched on in a previous section, the inductive transfer is a learning mechanism's ability to enhance task performance as a result of having learned a similar skill/task from another (i.e., earlier) one [2]. Hence, a general inductive-learning algorithm objective is to apply mapping based on training examples. So, for example, if the task is classification, a model thus learns to map class labels and input features, using seen and unseen data based on sets of assumptions concerning training data distribution. The assumption sets are referred to as inductive biases and include factors like search process and hypothesis space restrictions. These assumed biases limit the model's learning capacity but also streamline the process. In our proposed approach, we use transfer knowledge in model image classification related to the categories of "cat" and "dog", using a pre-trained model for detecting pristine images in comparison to forged images.

6.6.1 Datasets Environment

This experiment was conducted using multiple datasets [2]. The training used mainly dogs vs cats dataset which 25,000 images of dogs and cats in total each category have 12500 images. We can verify with the preceding output that we have 12,500 images for each category. Let's now build our smaller dataset, so that we have 3,000 images for training, 1,000 images for validation, and 1,000 images for our test dataset and that applied to both categories [7]. A collection dataset out of online and public existing datasets for training and testing. In total, we collected 1792 pair images with good quality to present different samples for copy-move forgery. The first one was constructed by Christlein et al [8], consisting of 48 base images and 87 copied with a total of copy-move forged images of 1392. The second database MICC-F600 was introduced by Amerini et al [9, 10] with 400 images. Caltech-101, image manipulation

dataset. IM dataset (240 images) [11]. The Oxford buildings dataset (198 images and 5062 resized images) [12]. Coverage dataset (200) [13] and collection of online and self-producer images. Note that, even the total images look bigger than what we used in training and testing, that is because we avoid using some images either because they are in bad shape or low resolution. Therefore, here we used selected ponch of images with total of 1348 image which devided for 1248 for training and 100 for testing task.

6.6.2 Implementation

Using Python 3.6, our experiments ran basic classification task on a network of similar images pairs in MatchNet. The model training data were kept in a secure H5 file, and the original convolutional neural network was built accord to [7]. Hence, the transfer learning network is more or less identical to the earlier network that employed pre-knowledge from the original CNN model. The towers have two main requirements, as follows. 1) Weight share: The towers have to be able to share weights, as this is how the network is able to learn image pairs simultaneously (i.e., running image pairs through the same weight set for every layer). 2) Towers as CNNs: Specifically, the towers must be CNNs of themselves, which then permits the learned weights from one network model to be used on the other. Therefore, the towers must have the same parameter and architecture names to enable weights to be transferable between them. This is accomplished as follows. In the weight-training stage, weights are initialized. At the completion of the training, the model is saved in an H5 file, which can then be loaded for additional training/testing utilizing the same weight parameters. The CNN networks comprise standard convolutions, normalization layers, ReLu and pool (i.e., frozen convolutional layers). However, they are unique in that images that are loaded are split, after which they are recombined prior to reaching fully connected

layers. Individual images are fed into input layers, with the outputs being vectors with representations comprising 4,096 elements. Next, the vector is fed to fully connected layer sets which have been re-trained to convert vectors into similarity/non-similarity scores. In the final step, the cosine similarity computes a Softmax loss of expected similarity vs. predicted similarity for image content. The total operation is functioning as deep learning classifier to classify the output in the designation target.

6.7 Experiment results

There are several tasks that can be evaluated using these features to compare to the state-of-the-art systems. While there are many such tasks in the unsupervised learning space, to bound the difficulty of evaluating the results of this work, the first set of experiments will simply attempt to test if the weights learned in the CNN are a better initialization for fine-tuning an image classification task than the provided weights which were learned on the training task. The rest of this section will outline in detail how these experiments were built and how weights were transferred. First, we present the result of training the original model to show how the results look like and then we'll use the new dataset to see how the knowledge transferred and show the result for the new task.

The baseline training model and the transfer learning model are layered in same architectures. The last layers (dense layers) were trained slightly different to serve the different output target. This architecture of layered network allowed us to employ the trained/or pre-trained network in the original model to extract the new features in the other model with the consideration of re-train the classifier to serve the new labeled case. Figure 6.2 shows the models summery in both tasks while table 6.1 shows the parameters' summary of the same model.

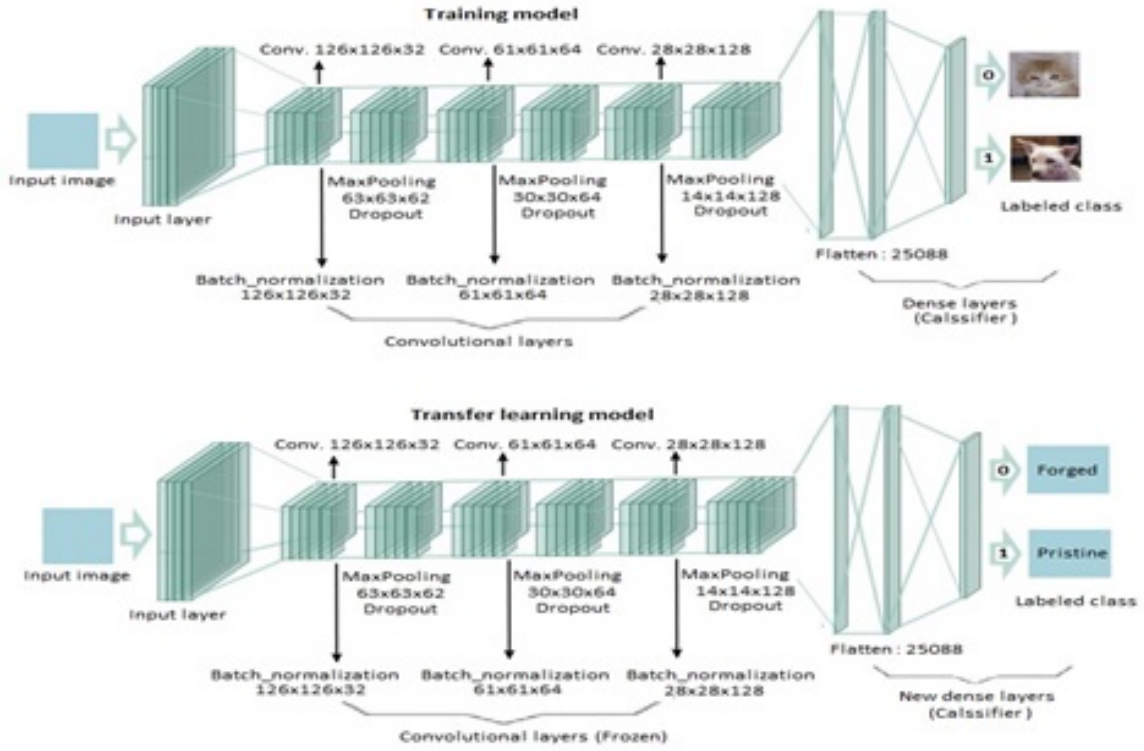

Figure 6.2: The summary of the processing of the transfer learning model form the original trained model

.

Table 6.1: Model summary information

Model parameters	Total parameters	Trainable parameters	Non-trainable parameters
Summary	12,942,786	12,941,314	1,472

In this work, the initial experiments aim to determine whether the weights that were learned in the CNNs improve the initialization of image classification task fine-tuning compared to the weights learned during the training task of the original dataset. The remainder of the section provides a detailed discussion on the construction of the experiments as well as the manner in which the weights are transferred during the tasks. Specifically, we will review and present the outcomes from the original model training, after which we will apply the new dataset in order to determine the means of knowledge transfer. The results of the new approach will also be provided.

6.7.1 Training data generator

In normal cases the all dataset images can't be uploaded to the process memory in one shout. Also, this may drop down the performance of the used GPU's / or CPU in the training task. Therefore, we will load the dataset in a group of 15 images at once in each time to all dataset using data generator. The other advantage of using data generator is to make many changes in each image which known as data augmentation to learn all possible image transformation to the used neural network after fixing the image scale according to the input layer which in our case is 126×126×32. Also this will present different image forgery techniques such as shearing, rotating, and scaling images. Figure 6.3 shows two examples of data generator. The output of this task can be summarized as following: Found 20000 images belonging to 2 classes. Training

Generator: Found 5000 images belonging to 2 classes. Found 0 images belonging to 0 classes. (See Appendix B has more figures)

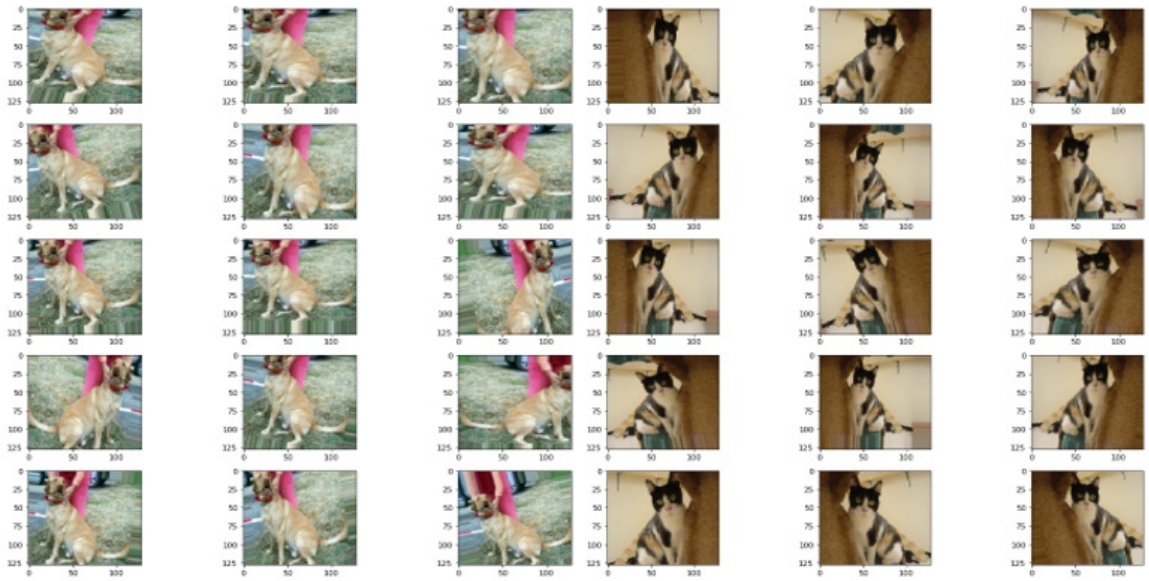

Figure 6.3: Examples of the data generator work using the original dataset

Note: Categories: 0: 'cat', 1: 'dog'. The original dataset has same number of the two categories that used in the training task as shown in the next illustration, figure 6.4.

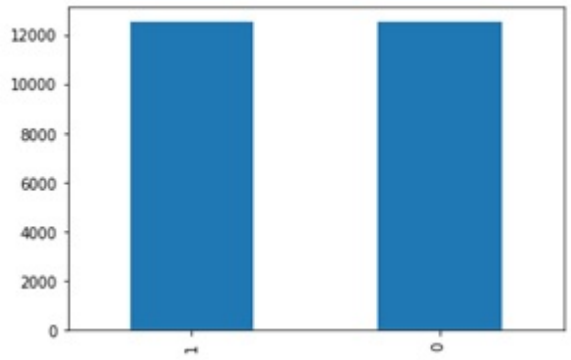

Figure 6.4: This figure presents the training result map to a dog is 1 and cat is 0

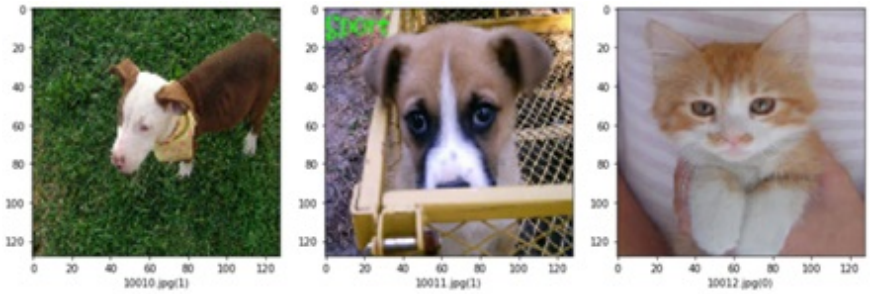

Figure 6.5: Random samples from the training result in the image's presentation

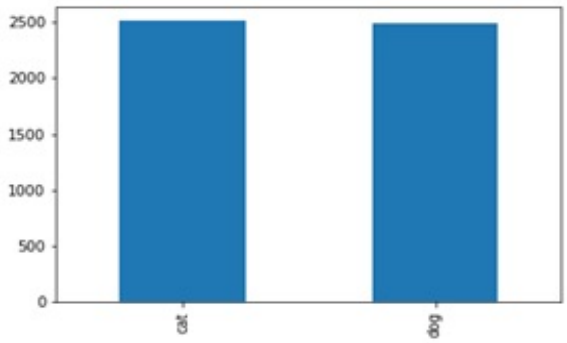

Figure 6.6: The validation result map to model using the original dataset: dog is 1 and cat is 0

6.7.2 Create Testing Generator

We will convert the predicted category back into our generator classes by using $train-generator.class-indices$. It is the classes that image generator map while converting data into computer vision

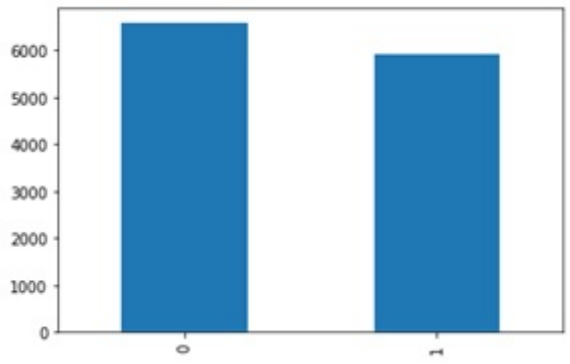

Figure 6.7: The testing result map to dog is 1 and cat is 0

Figure 6.8: The predicted result with images representation

Training task initial output: Found 998 images belonging to class 1, found 250 images belonging to class 0. After preparing the new dataset and go through the above steps we see the results:

Figure 6.9: Random sample images from the new dataset

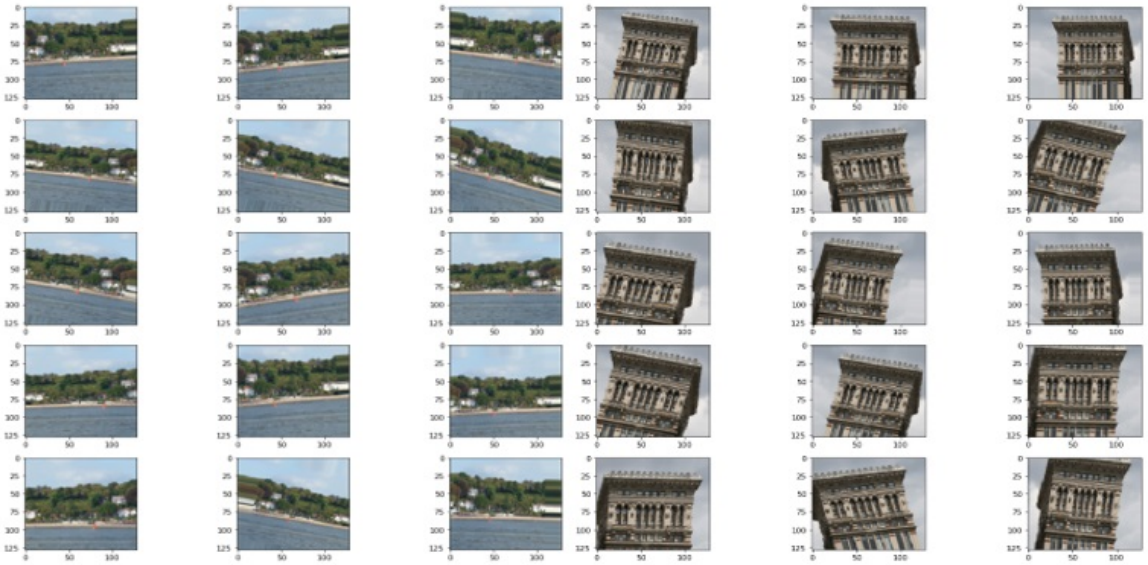

Figure 6.10: This figure shows examples of the data generator work using new dataset

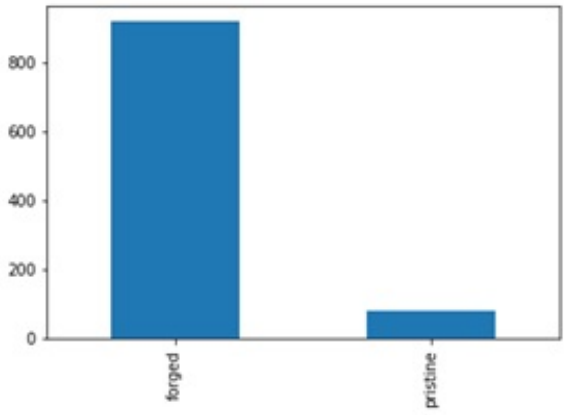

Figure 6.11: This figure shows a map result of training set for new dataset

Table 6.2 Numerical Training result which was presented in the above figure

Table 6.2: The numerical training result which was presented in the above figure

Image No.	Category	fillename	Image No.	Category	fillename
0	0	forged(1).png	1243	1	pristine(92).png
1	0	forged(10).png	1244	1	pristine(93).png
2	0	forged(100).png	1245	1	pristine(94).png
3	0	forged(1000).png	1246	1	pristine(95).png
4	0	forged(1001).png	1247	1	pristine(96).png

For the training task, we used the two categories with total images of 1248 images as we mentioned above. Here 6.12 we show random sample pairs images out of the used dataset:

Figure 6.12: Random samples of the two new categories with their labels

Now for the new dataset, the result should look like: ('dog': 1, 'cat': 0) => ('pristine': 1, 'forged': 0), i.e. each image labeled with a dog that means this image is a pristine image, and each image tagged in the cat is a forged image as you can see in figure 6.14.

.

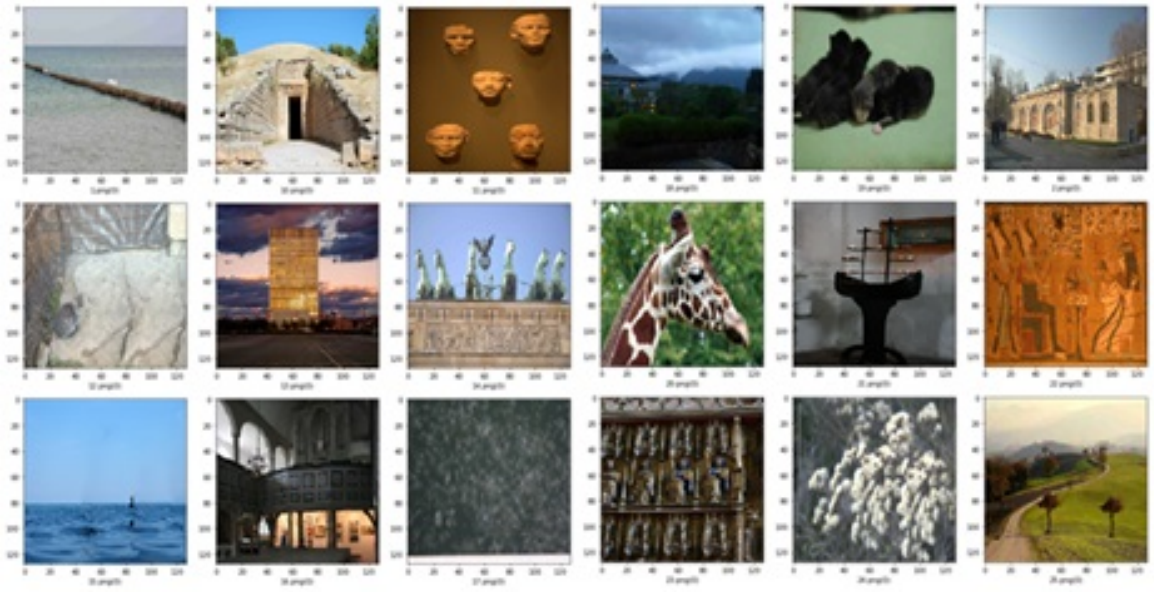

Figure 6.13: Predicted result with images using the new dataset

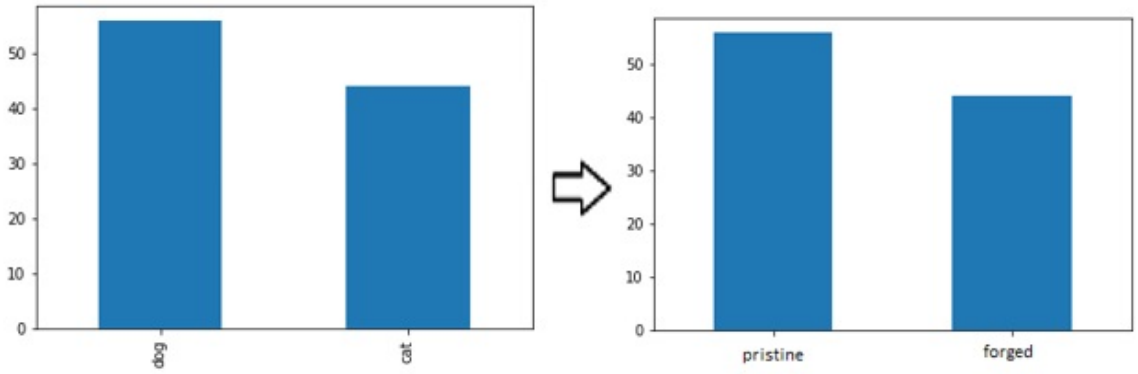

Figure 6.14: The image labels presentation for new dataset based on the original dataset labels

Based on the original model weights the second dataset will be judged. Next figure shows the deep learning classifier output after re-trained to deserve the new task. However, the output still shows some false detection which can be overtaken by using a closer model for the learning transfer assignment.

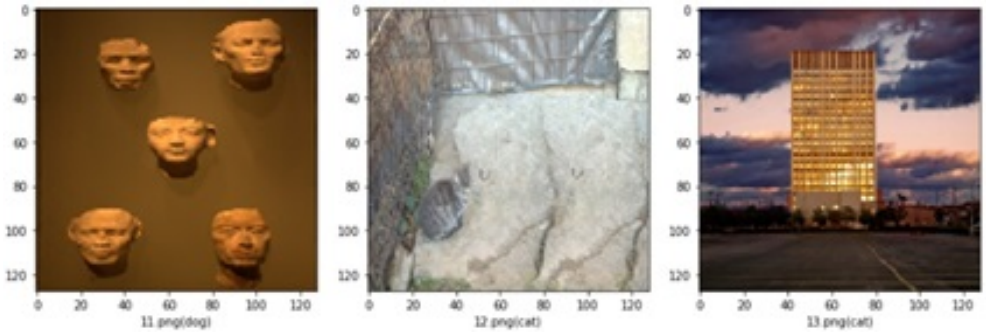

Figure 6.15: Evaluation of the testing result from the new dataset using the old categories labels for clear and fare judgement

.

According to the sample result presented in figure 6.15, the first image from the left hand is forged image and here shows as (dog==pristine). The other two images are (cat==forged) and that is correct. Now, in order to increase the accuracy of this model, we train the original dataset for a longer time using 25 Epochs with same conditions to avoid over fitting.

6.7.3 Validation and evaluation

The next figure 6.16 shows the validation data for both datasets before we flatten the data and feed our deep learning classifier.

As shown in the results, the validation accuracy of the baselines was higher than the accuracies using the weights learned in the original network. The next visual illustration shows clearly the gap between the training accuracy and validation accuracy for the both datasets.

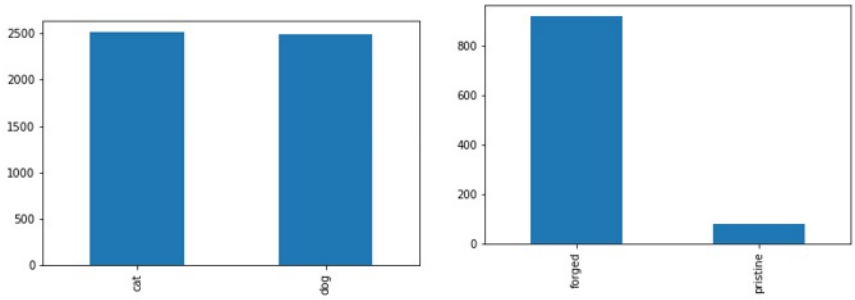

Figure 6.16: The validation data of the presented models in both datasets

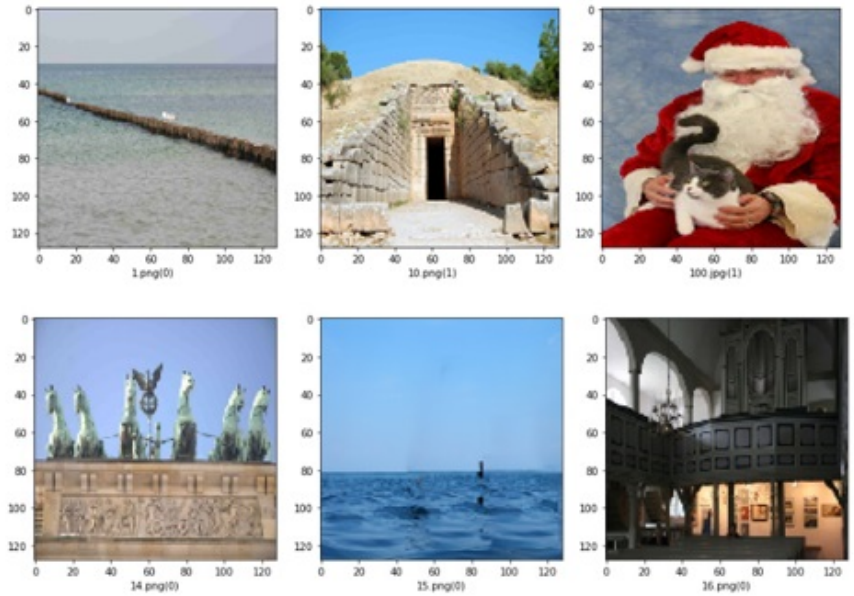

Figure 6.17: Some predicted results in images presentation for validation task

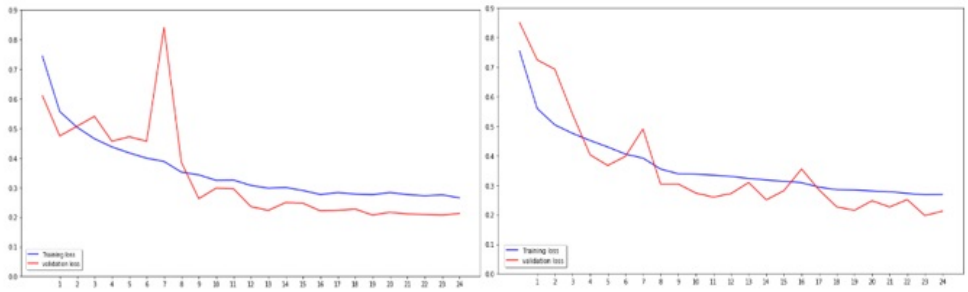

Figure 6.18: Training loss vs. validation loss in different training trial

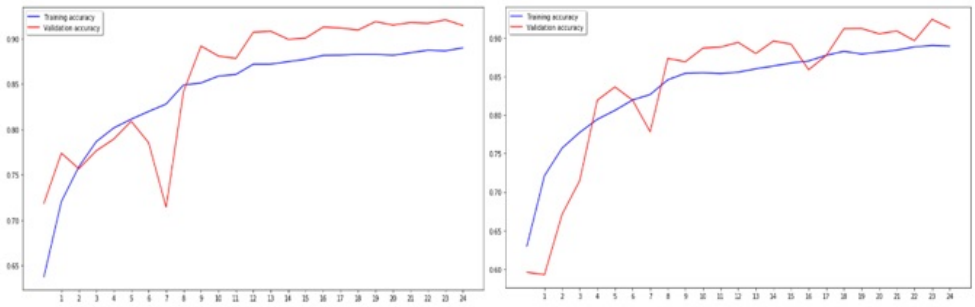

Figure 6.19: Training accuracy vs. validation accuracy in different training trial

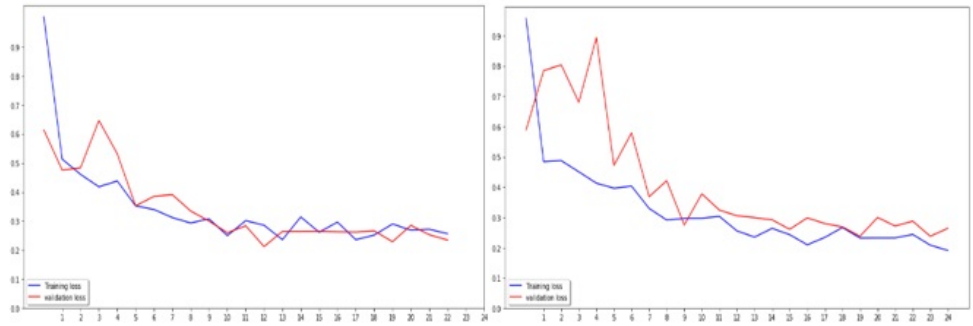

Figure 6.20: Training loss vs. validation loss in different training trial (new dataset)

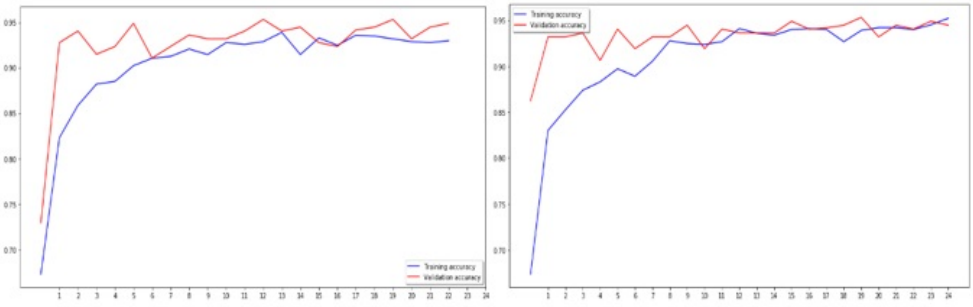

Figure 6.21: Training accuracy vs. validation accuracy in different training trial (new dataset)

From the figures 6.18, 6.19, 6.20 and 6.21 we can notes that the model is getting over fitting with new dataset, however, the model still able to maintain excellent validation accuracy. However, second training show more improvement in both loss and accuracy training vs. validations. Also, from the table 6.3 we can see that we achieved validation accuracy of 94.89% which 4% improvement in accuracy from the previous model.

Table 6.3: The loss and the validation accuracy of the both models

The model	F1-Score	Loss	Acc	Val-loss	Val-Acc.
Trained model	0.89	0.2683	88.93%	0.2122	91.29%
Transfer model	0.93	0.2583	92.94%	0.2347	94.89%

Evaluate this work with the-state-of-the-art is presented in the table 6.4.

Table 6.4: The evaluation based on the validation accuracy between two closer targets

The Model	Training data	Validation accuracy
[2]	MSCOCO	77%
	ImageNet subset	93%
Presented model	Dogs -and- Cats	91.29%
	Pristine -and- Forged	94.89%

6.8 Conclusion

The present work demonstrates that, by employing different CNN architectures, deep learning can be successfully applied in tasks such as image classification, image iden-

tification, and object recognition. Cost-effective image classification is achieved on manipulated and/or larger datasets, and improved image feature mapping are obtained from similar images in text metadata using CNNs. However, although using feature map representations is shown to be cheaper and faster, it does not improve the quality of the image classifications, indicating that this approach is not optimal for evaluating quality, given the weak correlation between feature labels and similar (and/or non-) images. Nevertheless, the results of the present work could lead to future investigations that include looking at other forms of forgery detection by applying the newly transfer learned weights. Overall, the present work indicates that metadata sampling and classification requires highly disciplined scaling model which can be scored by employing a pre-trained model with and that can be future extension steps to this work.

6.9 References

[1] R. T. C. Dipanjan Sarkar, Hands On Transfer Learning With Python, Packt, 2018.

[2] R. V. G. B. Soares, " Inductive Transfer," In: Sammut C., Webb G.I.(eds) Encyclopedia of Machine Learning. Springer, Boston, MA, 2011.

[3] E. S. Y. Jia, "Caffe: Convolutional Architecture for Fast Feature Embedding," arXiv, 2014.

[4] Im, D.J., Kim, C.D., Jiang, H., Memisevic, R.: Generative adversarial metric, 2016.

[5] T. A. B. X. Han, " MatchNet: Unifying Feature and Metric Learning for Patch-Based Matching.," Proceedings of Computer Vision and Pattern Recognition, 2015.

[6] D. Iter, "Image Classification using Transfer Learning from Siamese Networks based on Text Metadata Similarity," Stanford University, pp. 1 - 13, 2016.

[7] D. Sarkar, "A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning," Medium, 14 Nov. 2018.

[8] Jing, W., Hongbin, Z. “Exposing digital forgeries by detecting traces of image splicing,” in 8th International Conference on Signal Processing, vol. 2, IEEE, 2006.

[9] A. Mahendran, and A. Vedaldi, “Visualizing deep convolutional neural networks using natural pre-images,” International Journal of Computer Vision, 12(3): pp. 233-255, 2016.

[10] He, K., Zhang, X., Ren, S., Sun, J, “Identity mappings in deep residual networks,” In: arXiv preprint arXiv:1603.05027, 2016.

[11] Christlein, V., Riess, C., Jordan, J., Riess, C., Angelopoulou, E. “An evaluation of popular copy-move forgery detection approaches,” IEEE Transactions on Information Forensics and Security 7(6), pp. 1841-1854, 2012.

[12] James P., Relja A. and Andrew Z. [Online]. Available: http://robots.ox.ac.uk/ vgg/data/oxbuildings/. Acc. Jan. 2018.

[13] Im, D.J., Kim, C.D., Jiang, H., Memisevic, R.: Generative adversarial metric, 2016.

Chapter 7

Summary and Future Work

7.1 Summary of the thesis

Copy-move forgery detection (CMFD) had been widely adopted for use by people of all skill levels, due mainly to its user-friendly and ease-of-use approach. However, despite the relative simplicity of the strategy, there are still some challenges that make the outcome sometimes invalid or at least questionable. If a copy-move is performed by applying something in the image background to obscure evidence of forgery, this can be overcome by employing PatchMatch on the forged image's offset points. In this situation, the authentic image is needed to proceed with forgery detection, so a different method should be adopted. Our experiments indicate the presence of variance within the evaluations, which occurs also in identical images where there are alterations to the resolution or color, giving unequal F-scores. Despite these minor problems, the F-score generally exhibits optimal efficiency in the enhanced approach, showing FM = 0.98 and efficiency higher than 90.5.

In the present research, we tackle the problem of copy-move forgery in digital images. The literature review explored different methods that were reapplied, tested

and evaluated. Based on a comprehensive study of state-of-the-art approaches and the robustness and accuracy of existing methods and algorithms, we were able to choose the right path. Our enhanced PatchMatch algorithm, based on density, presented in chapter three, achieved high FM and was able to detect copy-move forgery in several different geometric phases by localizing the copy-move forgery based on offset points. The experiments on CMFD included RGB images and BW; in both cases, the F-score generally exhibited optimal efficiency in the enhanced approach and showed high value compared with the state-of-the-art techniques currently in use.

The problem of forged images has become a global phenomenon that is spreading mainly through social media. New technologies have provided both the means and the support for this phenomenon, but they are also enabling a targeted response to overcome it. Using the convolutional neural network (CNN) architecture approach shown in chapter four, to enhance a copy-move forgery detection efficiency was promising. Deep convolution learning algorithms are one such solution. These have been shown, in chapter five, to be highly effective in dealing even with image forgery that derived from generative adversarial networks (GANs).

The recent digital revolution has sparked a growing interest in applying convolutional neural networks (CNNs) and deep learning to the field of image forensics. The proposed methods aimed to train algorithms for solving a range of predetermined tasks. However, training a model that has been randomly initialized requires extensive time for computation as well as an enormous pool of training data to draw from. Moreover, such a model needs to be developed and redeveloped from the ground up if there are any alterations to the feature-space distribution. In addressing these problems, the transfer learning model in chapter six proposed a novel approach to training image forgery detection models.

In conclusion, to introduce new trends in the research field of image forgery detection,

we deeply looked at deep learning and neural networks to develop a new method that can improve the accuracy of copy-move forgery detection of digital images and that was achieved by the proposed CNN. The copy-move forgery detection and localization between the source and target areas were the main focus of this research, and the results of using the proposed CNN and GAN networks to implement that are highly promising. Even though datasets of copy-move forgery are available, but they are not sufficient for long-run training. Therefore, the proposed deep transfer learning can provide an option to overcome this problem.

7.2 Future work

For future work, we need to increase the use of deep learning to deal with image forgery in general and copy-move forgery in particular, especially in cases where the forgery industry is using the same strategy of using deep learning to introduce a new application for copy-move forgery. We also need to look at improving multi-copy-move forgery detection (MCMFD) algorithms to reduce the lack of detection using single copy-move forgery detection algorithms. These improvements can come by adopting the latest related technologies and components. It also will provide a tool capable of applying copy-move forgery detection in real-time situations. This could be even automated search application to detect the digital image forgery on media

Authors Publications

Journals Papers

1- Younis Abdalla, M. Tariq Iqbal, Mohamed Shehata, Copy-Move Forgery Detection and Localization Using Generative Adversarial Network and Convolutional Neural-Network, MDPI Journal on Information,Volume 10, Issue 9, 286, August 2019.

2- Younis Abdalla, M. Tariq Iqbal, Mohamed Shehata, Image Forgery Detection Based on Deep Transfer Learning, EJECE-European Journal of Electrical Engineering and Computer Science, Volume 3, Issue 5, 2019.

3- Younis Abdalla, M. Tariq Iqbal, Mohamed Shehata, Convolutional Neural Network for Copy-move Forgery Detection, MDPI Journal on Symmetry, Volume 11, Issue 10, 1280, October 2019.

4- Younis Abdalla, Mohammad Tariq Iqbal, and Mohamed Shehata, Fusion Approaches System of Copy-Move Forgery Detection, American Journal of Computer Science and Engineering Survey, Volume 6, Issue 1, 2018.

5- Younis E. Abdalla, M. Tariq Iqbal and M. Shehata, Copy-Move Forgery Detection Based on Enhanced Patch-Match, IJCSI International Journal of Computer Science Issues, Volume 14, Issue 6, November 2017.

Conferences Papers

1- Younis E. Abdalla, M. T. Iqbal, M. Shehata, Black Ice detection system using Kinect, presented at CCECE 2017, Windsor ON Canada.

2- Younis Abdalla, Tariq Iqbal, M. Shehata, Copy-Move Forgery Detection based on Enhanced PatchMatch, presented at 26th IEEE NECEC St. John's, Nov.15, 2017, St. John's, NL.

3- Younis Abdalla, Tariq Iqbal, Mohamed Shehata, Hand Gesture Detection and Pose Estimation Using Image Processing: A Survey, presented at 25th IEEE NECEC conference 2016, St. John's, NL.

4- Younis E. Abdalla, T. Iqbal, M. Shehata, Using MSER Algorithm to Characterize an Active Camera Movement, 24th IEEE NECEC conference Nov. 5, 2015, St. John's, NL.

5- Younis E. Abdalla, M. Shehata, T. Iqbal, Fall Detection – Vision-Based Indoor Environment, presented at the 23rd IEEE NECEC conference, November 3, 2014, St. John's, NL.

Appendix A

A.1 Appendix A

- raw data, extra images, extra spectra

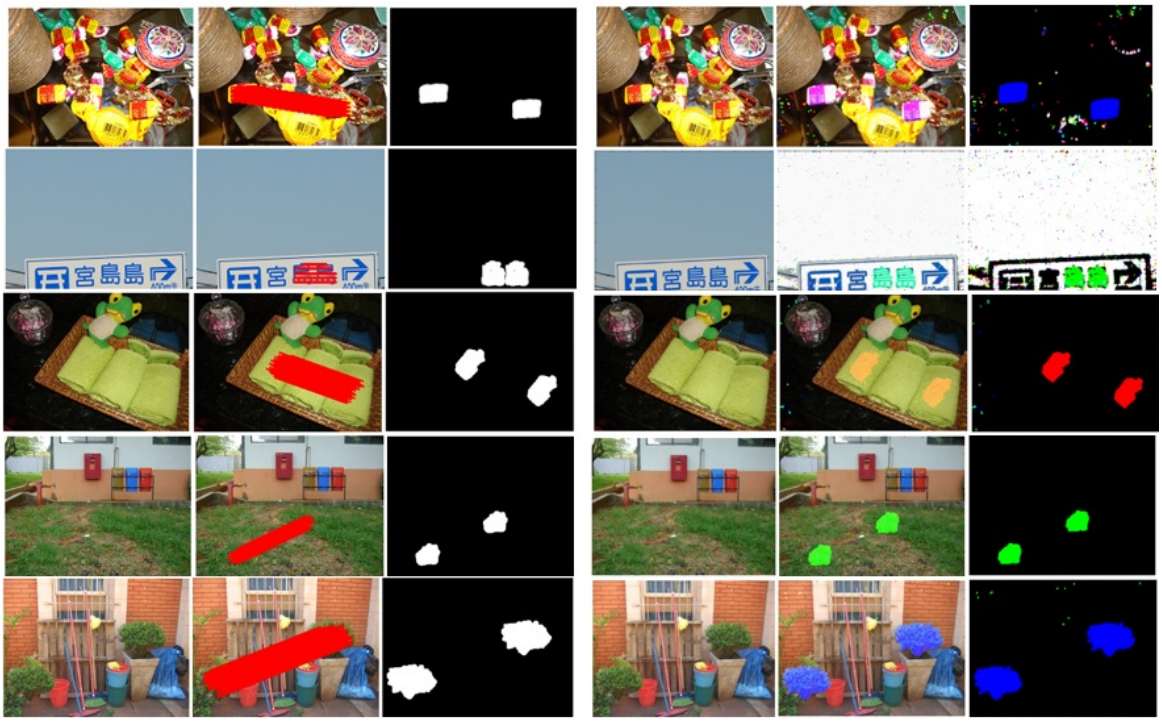

Figure A.1: Show the comparison between the (a) PatchMatch algorithm vs (b) DCT algorithm)

.

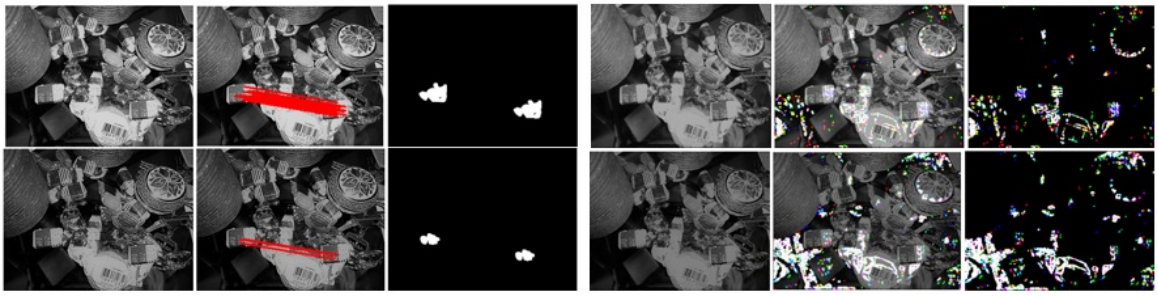

Figure A.2: Show the comparison between the (a) PatchMatch algorithm vs (b) DCT algorithm using gray images with different contrast)

.

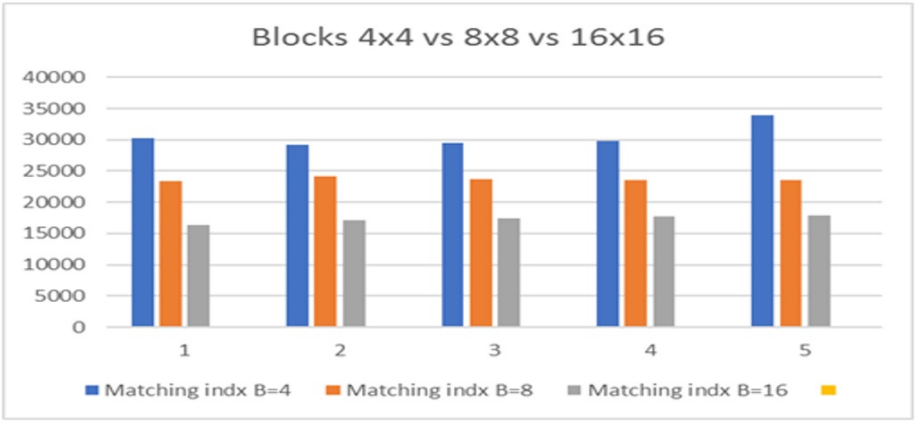

Figure A.3: Show the number of matching point in the same forged image based on the block size used for the scanning detection)

.

.

A.2 Appendix B

- raw data, extra images, extra spectra

.

Figure A.4: Show random result of the generator work original dataset

Figure A.5: Show random result of the generator work (new dataset)

.

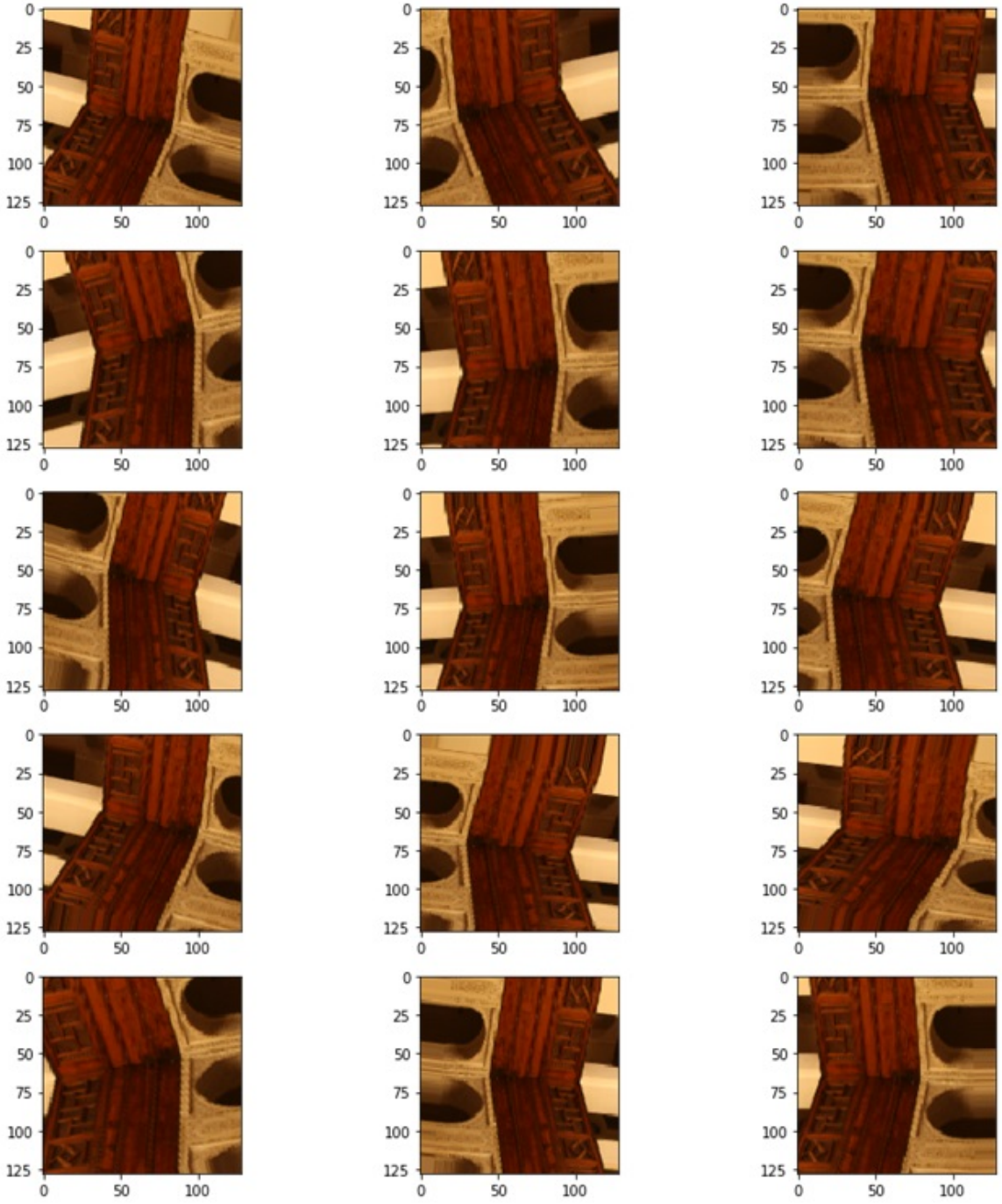

Figure A.6: Show random result of the generator work (new dataset)