Université de Montréal

Image Forgery Detection using Textural Features and Deep Learning

Par

Yishu Malhotra

Département d'informatique et de recherche opérationnelle
Faculté des Arts et des Sciences

Mémoire présenté à la Faculté des études supérieures et postdoctorales

en vue de l'obtention du grade de Maître ès sciences (M.Sc.)

en Informatique, option Intelligence Artificielle

Juin 2021

Université de Montréal

Département d’informatique et de recherche opérationnelle,

Faculté des Arts et des Sciences

Ce mémoire intitulé

Image Forgery Detection using

Textural Features and Deep Learning

Présenté par

Yishu Malhotra

A été évalué par un jury composé des personnes suivantes

Hafid Abdelhakim
Président-rapporteur

Esma Aïmeur
Directrice de recherche

Sébastien Roy
Membre du jury

Résumé

La croissance exponentielle et les progrès de la technologie ont rendu très pratique le partage de données visuelles, d'images et de données vidéo par le biais d’une vaste prépondérance de platesformes disponibles. Avec le développement rapide des technologies Internet et multimédia, l’efficacité de la gestion et du stockage, la rapidité de transmission et de partage, l'analyse en temps réel et le traitement des ressources multimédias numériques sont progressivement devenus un élément indispensable du travail et de la vie de nombreuses personnes. Sans aucun doute, une telle croissance technologique a rendu le forgeage de données visuelles relativement facile et réaliste sans laisser de traces évidentes. L'abus de ces données falsifiées peut tromper le public et répandre la désinformation parmi les masses.

Compte tenu des faits mentionnés ci-dessus, la criminalistique des images doit être utilisée pour authentifier et maintenir l'intégrité des données visuelles. Pour cela, nous proposons une technique de détection passive de falsification d'images basée sur les incohérences de texture et de bruit introduites dans une image du fait de l'opération de falsification.

De plus, le réseau de détection de falsification d'images (IFD-Net) proposé utilise une architecture basée sur un réseau de neurones à convolution (CNN) pour classer les images comme falsifiées ou vierges. Les motifs résiduels de texture et de bruit sont extraits des images à l'aide du motif binaire local (LBP) et du modèle Noiseprint. Les images classées comme forgées sont ensuite utilisées pour mener des expériences afin d'analyser les difficultés de localisation des pièces forgées dans ces images à l'aide de différents modèles de segmentation d'apprentissage en profondeur.

Les résultats expérimentaux montrent que l'IFD-Net fonctionne comme les autres méthodes de détection de falsification d'images sur l'ensemble de données CASIA v2.0. Les résultats discutent également des raisons des difficultés de segmentation des régions forgées dans les images du jeu de données CASIA v2.0.

Mots clés : Épissage d'images, Réseaux de neurones à convolution (CNN), ResNet-50, U-Net, Motif binaire local (LBP)

Abstract

The exponential growth and advancement of technology have made it quite convenient for people to share visual data, imagery, and video data through a vast preponderance of available platforms. With the rapid development of Internet and multimedia technologies, performing efficient storage and management, fast transmission and sharing, real-time analysis, and processing of digital media resources has gradually become an indispensable part of many people's work and life. Undoubtedly such technological growth has made forging visual data relatively easy and realistic without leaving any obvious visual clues. Abuse of such tampered data can deceive the public and spread misinformation amongst the masses. Considering the facts mentioned above, image forensics must be used to authenticate and maintain the integrity of visual data. For this purpose, we propose a passive image forgery detection technique based on textural and noise inconsistencies introduced in an image because of the tampering operation.

Moreover, the proposed *Image Forgery Detection Network (IFD-Net)* uses a *Convolution Neural Network (CNN)* based architecture to classify the images as forged or pristine. The textural and noise residual patterns are extracted from the images using *Local Binary Pattern (LBP)* and the *Noiseprint* model. The images classified as forged are then utilized to conduct experiments to analyze the difficulties in localizing the forged parts in these images using different deep learning segmentation models.

Experimental results show that both the IFD-Net perform like other image forgery detection methods on the CASIA v2.0 dataset. The results also discuss the reasons behind the difficulties in segmenting the forged regions in the images of the CASIA v2.0 dataset.

Keywords: Image Splicing, Convolution Neural Networks (CNN), ResNet-50, U-Net, Local Binary Pattern (LBP)

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AHE	Adaptive Histogram Equalization
BDCT	Block Discrete Cosine Transform
CFA	Color Filter Array
CMFD	Copy-Move Forgery Detection
CNN	Convolution Neural Network
DCT	Discrete Cosine Transform
DSIFT	Dense Scale-Invariant Feature Transform
DWT	Discrete Wavelet Transform
IWT	Integer Wavelet Transform
KNN	K-Nearest Neighbors
LBP	Local Binary Pattern
MLP	Multilayer Perceptron
PCA	Principal Components Analysis
SIFT	Scale-invariant Feature Transform
SVD	Singular Value Decomposition
SVM	Support Vector Machine

To my parents and siblings, for their endless love, support, and encouragement

and my friends Annanya, Ramya, Sravya and Siddu, who have been a source of inspiration...

Acknowledgments

I have received a great deal of support and assistance throughout the writing of this dissertation. I am incredibly grateful to my supervisor Professor Esma Aïmeur for her invaluable advice, continuous support, excellent supervision, and patience during my Master's. Her immense knowledge and great experience have encouraged me in my academic research and daily life. The door to her office was always open whenever I ran into a trouble spot or had a question about my research. She consistently allowed this thesis to be my work but steered me in the right direction whenever she thought I needed it. Without her guidance and continuous support, this dissertation would not have been possible.

I want to thank all the jury members, Professor Hafid Abdelhakim and Professor Sébastien Roy, for sparing their precious time to review and evaluate my thesis.

I thank my fellow lab mate Rim for teaching me several tips while writing my research work and pushing my limits. I would also like to thank Ramya and Sravya for being my colleagues and my sisters for being there for me whenever I needed them.

I want to thank my friend Siddu who always helped me solve the technical issues I encountered while working on my thesis.

I would also like to thank my friend Annanya for her continuous support and for always inspiring me to do better.

Finally, I would like to thank my parents and siblings for all the love and support they gave me to achieve my goals.

Chapter 1

Introduction

This chapter discusses the evolution of Image Forgery and the need for its detection to prevent or minimize the spread of fake news. The chapter is organized to include motivation factors responsible for my thesis dissertation, " Image Forgery Detection using Textural Features and Deep Learning." Furthermore, the chapter explains the problem statement, research objectives, and organization of my thesis dissertation.

1.1 Evolution of Image Forgery and motivation

Making a forged image is almost as old as photography itself. Photography quickly became the preferred means for creating portraits in its early years, and portrait photographers discovered that retouching their images to please the sitter may boost sales. With the advent of digital cameras and picture editing software, photo manipulation has grown more common. With the rapid development of Internet and multimedia technologies, performing efficient storage and management, fast transmission and sharing, real-time analysis, and processing of digital media resources has gradually become an indispensable part of many people's work and life. Due to the proliferation of easy-to-use and low-cost gadgets, digital visual media has become one of the most popular communication methods.

Furthermore, visual media have a greater expressive capacity than any other medium. It explains sophisticated scenes in a straightforward manner, which can be difficult to transcribe in different ways. While we may have historically had confidence in the integrity of this imagery, today's digital technology has begun to erode this trust. Digital Image Forgery is the intentional alteration of digital photographs to deceive to change public perception. The modification is carried out in such a way that it leaves no visible traces. From the tabloid magazines to the fashion industry and in mainstream media outlets, scientific journals, political campaigns, courtrooms, and the photo hoaxes that land in our e-mail in-boxes, doctored photographs appear with a growing frequency and sophistication.

The following are some of the most well-known examples of photo manipulation throughout history. We will concentrate on the cases that raise the most intriguing ethical questions or those that have been the most disputed or well-known. The photographers have also experimented with composition or *image splicing*, i.e., combining multiple images (or parts of images) into one composite. One of the earliest examples of composition is shown in Figure 1. General Sherman poses with his generals in this shot by renowned photographer Mathew Brady. General Francis P. Blair (far right) was added to the original photograph [1]. The picture on the right is from the same meeting, at which General Blair was not present.

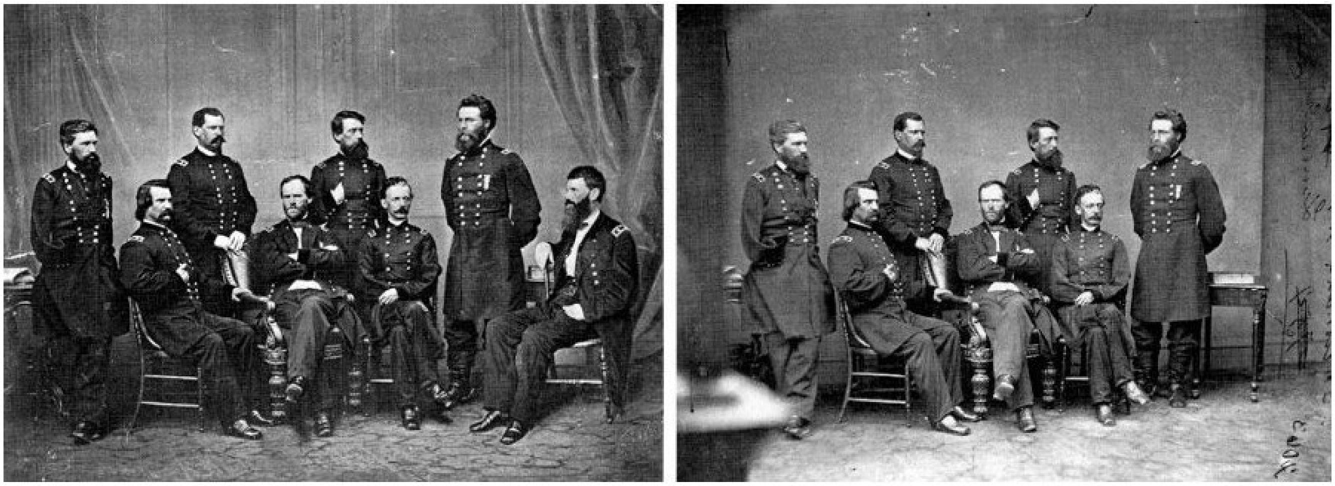


Figure 1: American general Francis P. Blair (right) was added to Mathew Brady's famous photo of General Sherman’s retinue because he was not at the meeting.

Such forged images are being used in today’s world to deceive the public and spread fake news. Therefore, image forensic tools must be developed so that image forgeries can be detected to prevent it.

1.2 Problem Statement

With the advent of technology and the introduction of low-cost, easy-to-use image editors like Adobe Photoshop, it is possible to splice and tamper with photos quite easily. Images can be modified to the point that it is impossible to tell the difference between an original and a forged image with the naked eye using such technologies. It becomes challenging to validate and maintain the integrity of photographs because of this. Such forged images are being used for deceiving people by propagating fake news and online deception. The field of digital forensics has emerged in recent years to assist in restoring some trust in digital photographs [2]. We propose a method that classifies images as fake or pristine, and we also analyze the obstacles in localizing the forged region in forged images.

1.3 Research objectives and our contribution

Digital picture manipulation is no longer limited to specialists with the introduction and widespread availability of useful picture editing tools and software. Sumopaint and Photoshop CC are some of the most well-known picture editing applications available online. Manipulation of visual media is no longer an arduous task with such readily available technologies [3]. This jeopardizes image credibility and undermines public trust in social media and social media communication. Image forgery detection and localization have become necessary to validate and protect the integrity of images, which is addressed in this thesis. In this thesis, we try to answer the following research queries:

- What is an alternative to the existing solutions and practical approach for image forgery detection?
- How can textural inconsistencies in the image content be used to detect image forgeries?
- Why currently available deep learning segmentation models are unable to localize the forged areas in these images?

We tried to answer the above-mentioned research queries by our contribution, as explained below:

- We extracted the textural and noise residual patterns of the images in the CASIA v2.0 dataset.
- We evaluated the effectiveness of different image representation approaches on forgery detection accuracy.
- We evaluated our proposed approaches on the CASIA v2.0 dataset
- We ran experiments on various sophisticated deep learning architectures and techniques to analyze the difficulties they face in localizing the forged regions.
- Finally, we captured the experiment results, graphs and discussed the results.

1.4 Thesis Organization

The thesis is organized into five chapters, including the introduction chapter.

- **Chapter 2** provides an insight into the literature review of the recent research work in image forgery detection. We discuss and compare the various types of image forgery detection techniques along with their limitations.
- **Chapter 3** addresses the limitations discussed in Chapter 2 and provides an insight into our proposed method for image forgery detection.
- **Chapter 4** discusses the captured results after running experiments on the CASIA v2.0 dataset.
- In **Chapter 5**, we finally conclude the thesis along with the future work.

Chapter 2

Literature Review

In this chapter, we discuss recent works that have been done in the areas related to this thesis.

2.1 Image Forgery

Forgeries of any kind are usually done with the intent to deceive others. Image forgery means manipulating the digital image to conceal some meaningful or valuable information of the image [4]. One of the main reasons behind image forgery is an intent to deceive others for altering public perception. In today's world, where we have access to such sharp and high-quality images, images can portray a tremendous amount of information. Thus, image forgery techniques are used for manipulating the information provided in an image. With the advent of inexpensive and easy-to-use image editors such as Adobe Photoshop, Corel Draw, Pixlr, it has become easy to manipulate images wrongfully. With the help of such tools, images can be manipulated to such a degree that one cannot differentiate between an original image and a manipulated image from one's naked eye. This makes it very difficult to authenticate and maintain the integrity of images.

Deception is defined as a message knowingly transmitted by a sender to foster a false belief or conclusion. In today's world, image forgery is also being used to propagate online deception amongst the users of various social media platforms. Surprisingly, image forgery and deception are not new. It has been recorded in history in the early 1840s. Hippolyte Bayard, an early inventor of photographic processes, is the first to create a fake image. Bayard's image is the first known instance of a faked photograph taken just one year after what can be considered photography's official "start" date in 1839. Bayard created the first staged photograph entitled "Self Portrait as a Drowned Man." He pretends to have committed suicide in the image sitting and leaning to the right (Figure 2) [5]. Since then, the number of fake images has grown drastically and is getting difficult to identify.

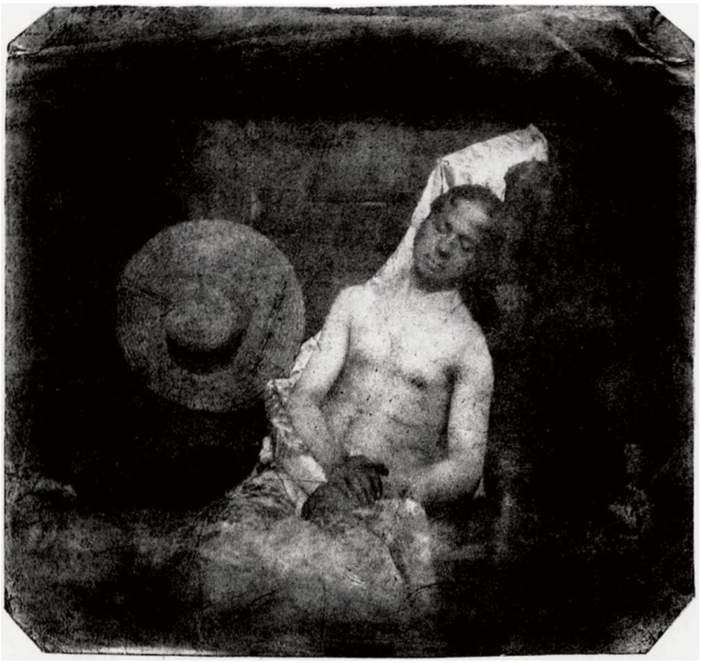


Figure 2: Self-portrait as a drowned man by Hippolyte Bayard *[5]*

This has led to the decrease in the reliability of digital images and escalated copyright issues leading to the necessity of image authentication. Thus, in today's world, detecting fake images has become necessary to prevent fake news, misinformation, and online deception. But before understanding the techniques to detect forged images, it is essential to understand the different types of image forgery techniques and how they are implemented to produce forged images.

2.2 Types of Image Forgery

In an era filled with technological advancements, an image can be forged in many ways. Amongst them, the most common image forgery techniques are:

- *Copy-move* / Cloning / Region Duplication Forgery
- Image Splicing / *Photomontage*
- *Image Retouching*

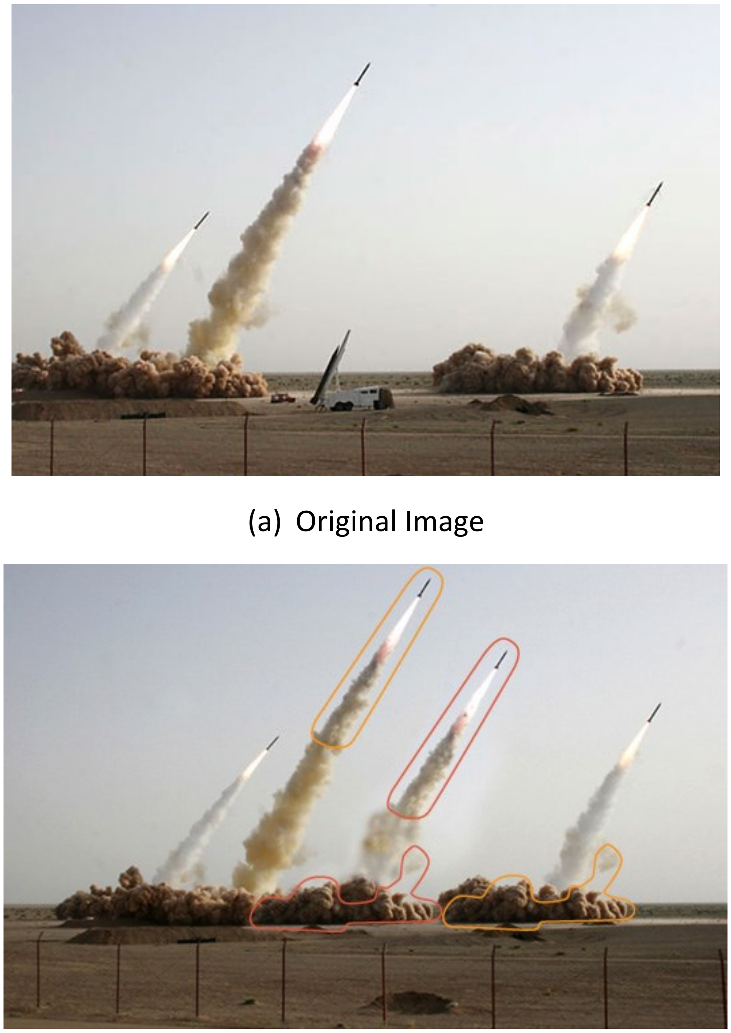
These three techniques are the most highly used techniques to forge images to deceive people and spread misinformation [6].

2.2.1 Copy-Move Forgery

The copy-move image forgery technique is one of the most popular image forgery techniques because it is simple, easily implemented, and effective [7]. A copy-move image forgery is performed by copying certain parts of a given image, moving them to the desired location, and pasting them in the same image. Such kinds of forgeries are done to either highlight a particular object in an image or to conceal an element in it.

As both the source and target region originate from the same image in copy-move forgery, properties such as dynamic range, Illumination conditions, noise, and color temperature are usually well matched between the forged region and the remainder of the image [8]. One of the ideal regions for copy-move forgery in an image is the textured region. Textured areas have identical noise variation and color properties regarding images, making them unperceivable for human vision and making it even more challenging to detect forged and inconsistent regions in an image [9]. In some situations, more processes such as noise addition, blurring, rotation, and scaling are also carried out on the forged image to make these copy-move operations look more natural. This makes it even more challenging to detect the forged region in the image through human eyes. Since the copied parts in an image can be of any shape and location, it becomes computationally impossible to search for all the possible image locations and sizes [2].

Figure 3 shows an example of an image forged using the copy-move forgery technique. Iran released the forged image, and the entire western media published it, including BBC News, The Los Angeles Times, and The New York Times [10].



(a) Original Image

(b) Copy-Paste Forged Image

Figure 3: An Example of Copy-Move Forgery *[10]*

In addition to copy-move forgery, sometimes the copied part belongs to a separate image, resulting in a spliced image. This technique will be discussed in detail in the next section.

2.2.2 Image Splicing

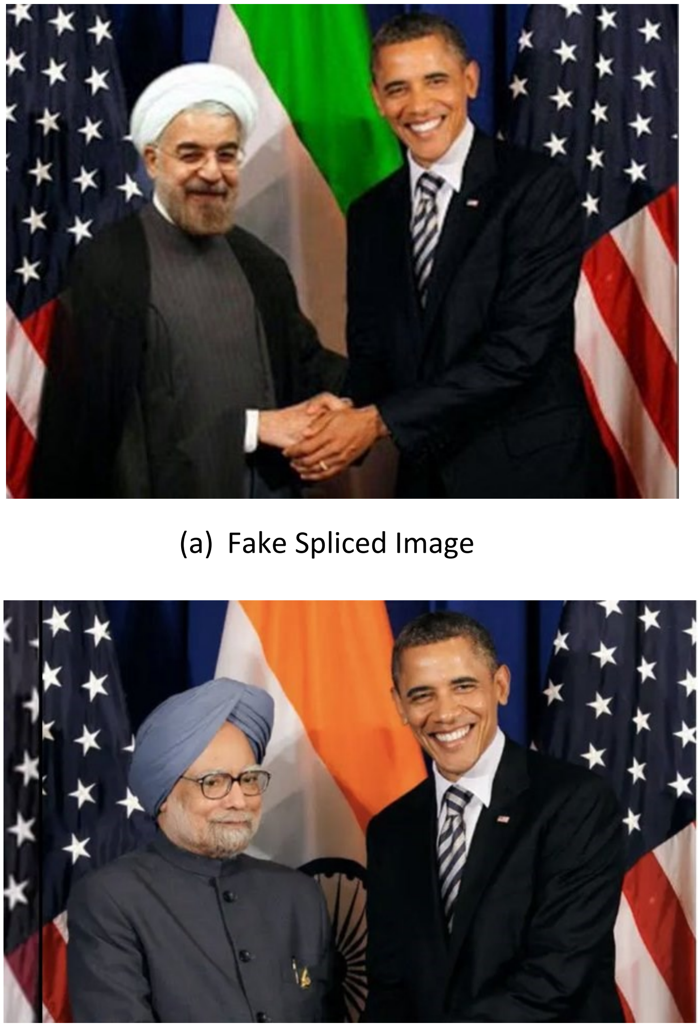
Forging images is not limited to copying parts of an image and pasting them in the same image. In many cases, two or more images might be used to create a forged image. A common form of photographic manipulation is the digital splicing of two or more images into a single composite [2]. Splicing images carefully can output realistic images to such an extent that the border of the spliced regions can be imperceptible to the human eye. This forgery technique is more aggressive than both copy-move forgery and image retouching. Digital image tools such as Adobe Photoshop and Corel Draw, which are simple to use and are readily available, can be used for splicing images together.

Several infamous news reports involve the use of spliced images. One of the most recent and notorious cases of a spliced images that surfaced on social media was former US President Barack Obama shaking hands with the Iranian president Hassan Rouhani [11]. The image was a tweet by Congressman Paul Gosar on January 6, 2020, with the description, “The world is a better place without these guys in power.” It was then opposed by other political leaders stating that it was a fake image. The original image involved former Indian prime minister Manmohan Singh shaking hands with former US President Barack Obama. This image was spliced with the Iranian president’s photo and tweeted, as seen in Figure 4.

Image splicing usually leads to a change in texture and background color composition, leading to local noise variances. For an un-tampered natural image, the noise variances across different regions typically differ only slightly. But with spliced regions from another image with a significantly different intrinsic noise variance, the inconsistency of local noise variances becomes telltale evidence of tampering [12]. Hence, one way of detecting such a kind of forgery is by checking the variation of the component, such as texture, dynamic range, and color palette. [13]. Apart from copy-move and image splicing, the third type of forgery does not perform these operations and instead focuses on enhancing specific features of the image. It is known as Image Retouching, and the next section discusses it in detail.

2.2.3 Image Retouching

Image retouching does not involve operations such as copying and pasting a region in an image. Instead, it enhances certain features of an image, such as color and image brightness. Such operations may be carried out to improve or degrade an image so that it looks more pleasing to the eye. Image retouching is popularly used in newspapers, magazines, and films. It is a passive image forgery technique and does not significantly impact the image [13]. But it must be noted that it is a type of image manipulation and thus can be characterized as an image forgery technique.



(a) Fake Spliced Image

(b) Original Image

Figure 4: Fake Spliced Image and the Original Image *[14]*

Usually, image retouching enhances certain features of an image to make the image look more appealing to the masses. These enhancements in an image are ethically wrong, mainly when they are being used to deceive people. Such retouching of images can be primarily seen in advertisements where the companies want their products to look more appealing to the public so that they are attracted to buy their products. It has been often seen that celebrities influence their fans by posting their photos on social media, which depict a particular physique, skin tone, etc. Many of these photos have undergone image retouching to make them more appealing.

Image retouching can be divided into two subcategories as follows:

- *Technical Retouching*
- *Creative Retouching*

Technical retouching is used in case of image restoration or enhancement. It deals with adjusting noise, colors, white balance, sharpness, tonality, and visible flaws in the image [15].

Creative retouching is a technique adopted for commercial use or as a form of art to make the images sleeker and more interesting for advertisements and art [16].

Based on the intent and the application, some image manipulations may be considered an art form as they may involve creating new images that do not deceive people. An example of image retouching can be seen in Figure 5.

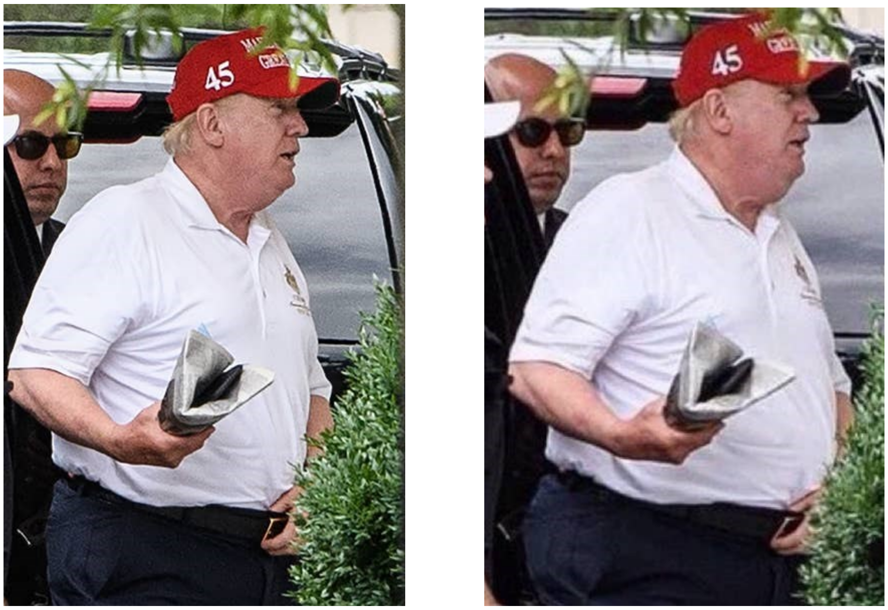


Figure 5: The image on the left is an original image, and the image on the right is retouched *[17]*

In the above figure, the image on the left is the original image of former US President Donald Trump, and the image on the right is his retouched image to make him look fatter. Such retouched photos can aid the spread of fake news and can influence public opinion. Thus, image forensics is required in today's world to detect such forged images and bring forward the truth behind them.

2.3 Image Forgery Detection

In today's world, social media plays an integral role in a person's day-to-day life. Many people use various social media platforms such as Facebook, WhatsApp, Snapchat, Instagram, etc., to share images, text, and videos. Furthermore, the evolution of mobile devices has allowed people to capture images anywhere and share them on various social media platforms that very moment. Thus, images have become one of the most highly shared media types on social media platforms [18]. It makes it necessary to monitor the images that are being shared on various social media platforms.

For many years photographs have been used as a part of the evidence in various judicial courts. Even though such analog pictures can be used to create composites, it requires immense knowledge and is quite time-consuming compared with digital images [19]. However, with the advent of readily available and powerful image editing tools manipulating images has become relatively straightforward.

The increase in the available digital image data has bolstered the growth in available powerful image editing tools. It allows people to modify and manipulate images effortlessly, threatens images' credibility, and decreases the public's confidence in social media and social media communication. Thus, image forgery detection has become the need of the hour to authenticate and maintain the integrity of images.

Unlike traditional semantic object detection, image forgery detection gives more importance to manipulating artifacts than the image content, making it necessary for image forgery detection techniques to learn rich features [20]. Figure 6 provides a broad classification of the various image forgery detection techniques.

Image forgery detection techniques can be classified into the following two categories [13, 21, 19, 7, 22, 23]:

- *Active Approach*
- *Passive / Blind Approach*

In the active approach of image forgery detection, first, the pre-processing of the image is carried out, followed by embedding a cipher key in the image. This key can then be used at the receiving end to authenticate the image [24]. In the active approach for image forgery detection, this cipher key embedded in the image is imperative for authentication and integrity checks. At the time of generation of the image, some code or ciphertext is embedded in the image using the active approach, which is usually imperceivable to the human eye. Therefore, it can detect if an image has been forged if the embedded code or the ciphertext cannot be extracted from the image. However, this may require dedicated hardware or software to embed such information in an image.

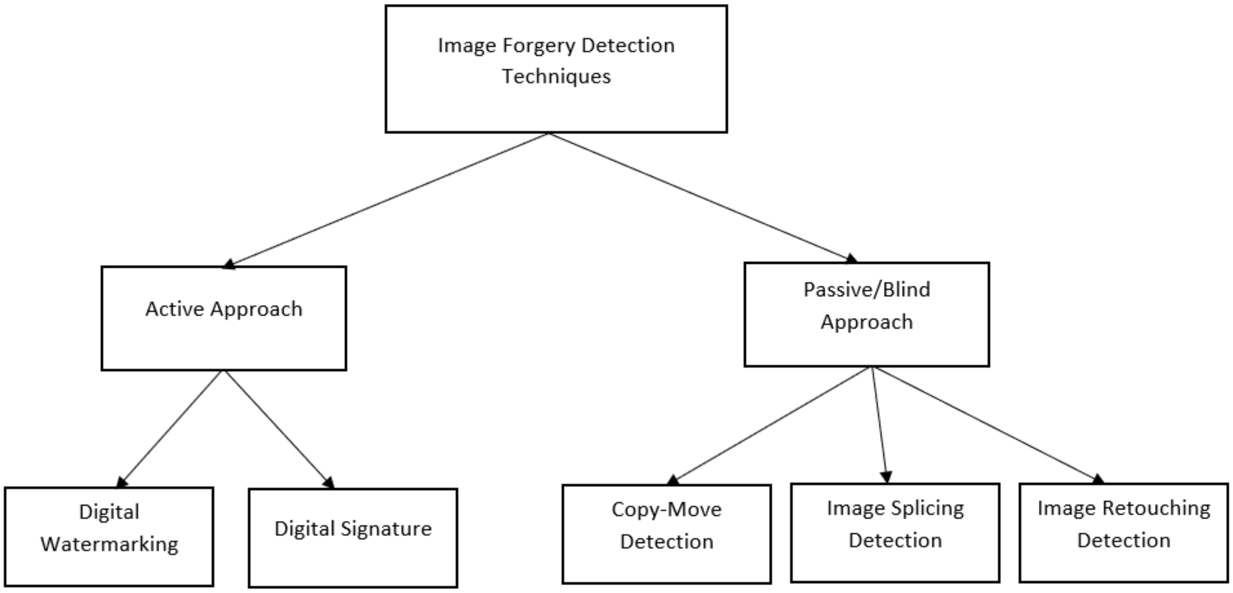


Figure 6: Classification of Image Forgery Detection Techniques

The active approach can be mainly categorized into two types:

- *Digital Watermarking*
- *Digital Signatures*

Digital watermarking is an active approach to image forgery detection, which involves embedding a security structure into the image. A *digital watermark* is an identification code permanently embedded into digital data, which can carry information about the copyright owner, the creator, the authorized consumer, etc. [25].

Digital signatures are used to sign images using a privately held decryption key. This signature can then be authenticated using the corresponding publicly revealed encryption key [26]. Usually, it is difficult to forge the digital signatures embedded in an image.

On the other hand, the passive image detection approach does not involve any pre-processing. Instead, it analyzes the raw image data for semantic and statistical inconsistencies in the image content to detect and localize image forgeries. In contrast to the active approach, the passive approach does not require any previous information about the image. Instead, it takes advantage of specific detectable changes that forgeries can bring into the image [27]. The passive image forgery detection techniques do not require any prior information about the input image to detect image forgery. Instead, these techniques detect forgery based on the disturbances in the intrinsic features of the image that might have been introduced during its manipulation process. The images downloaded from the Internet have no prior information. Hence the active forgery detection techniques are of no use for such kinds of forged images. Therefore, it is evident that passive forgery detection techniques are comparatively more practical today.

The passive approach can be further divided into two categories forgery-type dependent and independent detection techniques. The forgery-type dependent detection techniques are designed for specific forgeries, such as copy-move or image splicing. In contrast, the independent techniques are designed to detect forgeries regardless of the type of forgery. The independent techniques exploit three different artifacts to detect general tampering: traces of re-sampling, compression, and inconsistencies. The forgery type-dependent techniques can be divided into two categories: copy-move detection techniques (single image-based forgery) and image splicing techniques (multiple image-based forgeries) [27]. All these image forgery detection techniques are discussed in detail in the following sections.

2.3.1 Digital Watermarking

Digital watermarking can detect image forgery using the security structure that it embeds in an image. We can evaluate the integrity of an image using this embedded security structure. If any inconsistencies or discrepancies are found in this security structure embedded in the image, we can conclude that the image has been forged or manipulated [21]. Also, by doing the reverse analysis of the security structure, we can locate the forged or manipulated region in the image.

Digital watermarking can be divided into three branches:

- *Robust*
- *Fragile*
- *Semi-Fragile*

Robust watermarks are generally used for the task of copyright protection. Even though these watermarks provide excellent robustness and transparency, they cannot determine if a digital image has been forged and cannot localize the forged regions within the image [28]. Such watermarks are used in IPR protection applications as they are designed to resist host signal manipulations. Unfortunately, none of the available watermarking schemes can practically resist every type of modification, regardless of their austerity. Thus, it can be said that robustness refers to a specific degree of host signal depreciation and a subset of all the possible manipulations [29].

Watermarks designed to be vulnerable to all modifications so that they are undetectable by the slightest host data manipulation are called Fragile watermarks. Fragile watermarks can be designed more easily than robust watermarks, and thus they are commonly used in authentication services [29]. Fragile watermarks are made to detect even the tiniest variations in pixel values. Fragile watermarks treat digital images as an entirety and prohibit any alteration or manipulation. Thus, the digital image cannot pass the certification mechanism even if there is only a tiny alteration.

The Semi-Fragile watermark class provides selective robustness to a set of modifications deemed allowable and legitimate while remaining vulnerable (fragile) to all others. These watermarks can also be utilized instead of fragile watermarks to provide authentication. The semi-fragile watermark, which combines the benefits of both the robust and fragile watermarks, is primarily used for fuzzy digital image authentication. Additionally, semi-fragile watermarking technology can locate and even recover tampered locations [28].

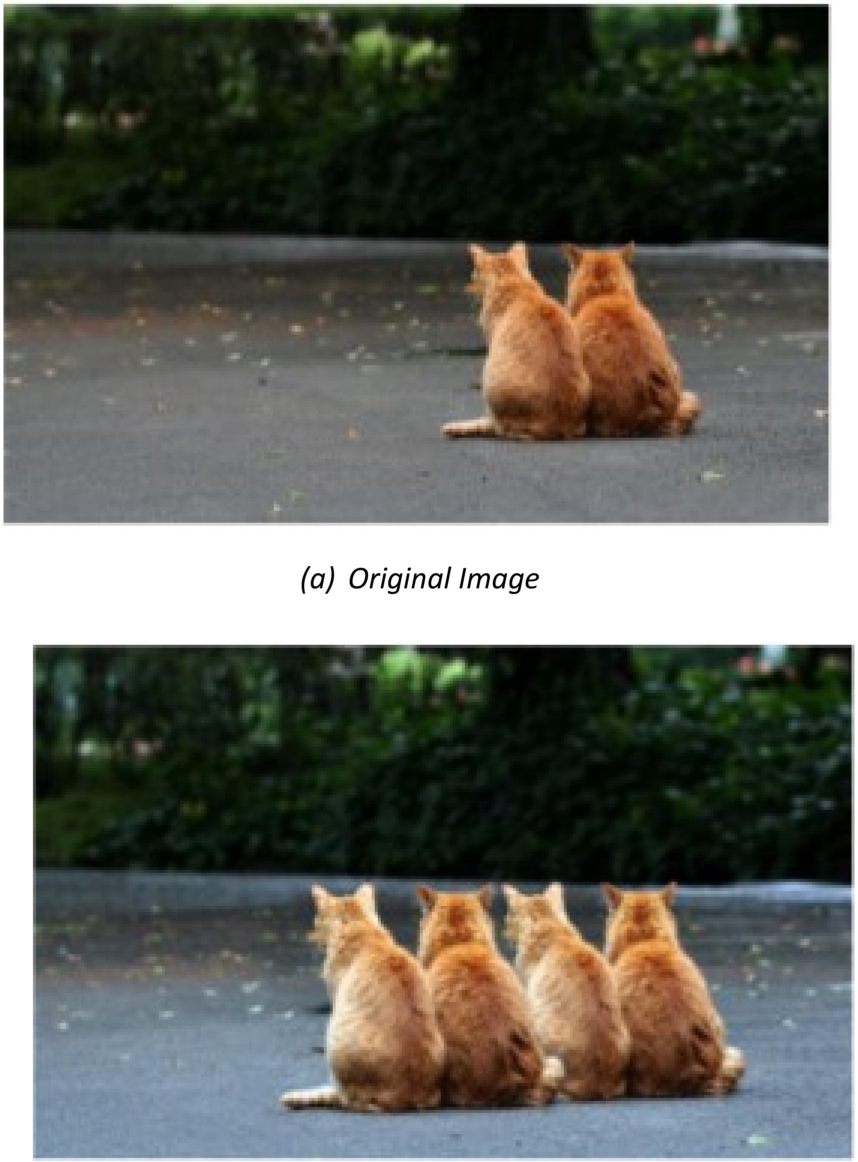
However, in the last few years, the use of passive forgery detection techniques has grown rapidly. Researchers have shown more interest in passive forgery detection techniques, specifically those dealing with detecting copy-move and image splicing forgeries. The following sections discuss such forgery detection techniques in detail.

2.3.2 Copy-Move Detection

Copy-Move manipulation is used to make an object 'disappear' from the original image by covering it with a small fragment copied from another portion of the image. This method can also be used to duplicate objects that already exist in the image. Because these cloned blocks are made from the same image, their features will blend in with the rest of the content, making it difficult for the human eye to notice them. Copy-move forgery detection is a passive or blind method of detecting image manipulation in which one or more sections are transcribed and pasted within the same image [4]. When the duplicated section is relocated, it is sometimes followed by a blurring effect applied to the modified region's boundaries to reduce the irregularities between the original and manipulated areas [63]. Although it may be simple to spot the copied element, geometric operations and post-image processing operations make it difficult for forgery detection techniques. Post-processing operations and intermediate image processing operations are the two types of operations that are usually used. For structural harmonization and correlation between the copied region and the target image, intermediate operations are used. Mirroring, scaling, rotation, illumination or chrominance modification, and other operations are examples of these operations. Intermediate processing can be used in the middle of two or more operations in the practice of forging. Whereas to hide perceptible touches in the image, post-processing operations such as blurring or additive noise and JPEG compression are used [4].

The search for duplicate areas is the main emphasis of detection algorithms for this type of modification. When combined with additional post-processing techniques, such as geometrical transformations or the application of color filters, it can make detection by existing methods rather difficult [30]. Figure 7 depicts the use of the copy-move forgery technique. Figure 7 (a) shows an original image with two cats, whereas Figure 7 (b) is a copy-move forged image.

Two equal portions based on the attributes of the blocks into which the image is divided are the most common evidence used to detect copy-move operations.



(a) Original Image

(b) Copy-Move Forged Imaged

Figure 7: Copy-Move Forgery Example *[30]*

There are three types of detection strategies for copy-move forgeries:

- *Block-based*
- *Keypoint-based*
- *Deep Leaning-based*

Various researchers in the past years have developed many block-based strategies. A block-based strategy was proposed by Fridrich *et al.* [9] in 2003. It used the *Discrete Cosine Transform (DCT)* coefficient characteristics from overlapping picture blocks as one of the initial approximations to locate copied areas inside images. It was one of the first methods to employ DCT for detecting copy-move forged images. Popescu *et al.* [31] introduced a method for detecting duplicate parts in a digital image. Instead of DCT, their technique used *Principal Components Analysis (PCA)*. The approach performed PCA on small fixed-size image blocks before lexicographically sorting each block. This methodology demonstrated a high degree of efficiency in detecting copy-move forgeries and the ability to detect them even in the presence of considerable amounts of corrupting noise. Cozzolino *et al.* [32] presented a method that combined dense-field approaches with *Zernike moments*. To distinguish the manipulated areas in a digital image, the use of Singular Value Decomposition (SVD) was proposed in [33]. The use of lexicographic classification helped to identify similar blocks. This method proved to be both reliable and effective. For tampered images subjected to Gaussian blur filters, noise contamination, and compressions, the experimental results indicate the validity of this technique. Mahmood *et al.* [60] suggested a new method for detecting copy-move forgeries using the concept of stationary wavelets. They reduce the feature dimension using DCT in this approach.

In 2009, Huang *et al.* used the *Scale-invariant Feature Transform (SIFT)* algorithm to identify copy-move forgery in digital photos [34]. The SIFT computation algorithm using the block matching function was presented by the authors. Even when the images are noisy or compressed, this technique produced good results. To identify copy-move forgeries, Khayeat *et al.* introduced the enhanced dense scale-invariant feature transform (DSIFT) descriptor [35]. They also presented neighborhood clustering to eliminate false matches. They improved the DSIFT descriptor by first identifying the prevailing orientation using second and third-order central moments. Then, they took a circular region rather than a square one to reduce border effects. This method is somewhat robust to post-processing operations such as rotations. The authors of [36] suggested a strategy based on *Speeded Up Robust Features (SURF)*, which has superior key point characteristics than SIFT since it works better with postprocessing techniques like blur variations and brightness.

The approaches based on key points, on the other hand, have a visual output problem since the copied and pasted regions are made up of lines and points that do not have an intuitive and clear visual effect. Amerini *et al.* presented an approach based on SIFT [37]. Duplicated portions in images can be detected using this method. In addition, this method also determined which geometric transformation was used. This method can also be applied to compressed images with a poor quality factor. Muhammad *et al.* presented a *Dyadic Wavelet Transform (DyWT)* based blind copy-move forgery detection approach [38]. The method primarily relied on two types of information: noise inconsistencies between image blocks and the similarity between these blocks. The authors carried out the experiments in three scenarios: i) copy-move region without rotation in same size images, ii) copy-move region with and without rotation in different size images, and iii) images with different Quality (Q) factors. These experiments showed the superior performance of this method compared with some of the previously mentioned methods. Zhao *et al.* presented a method for analyzing and detecting duplicate regions in images based on discrete cosine transform (DCT) and singular value decomposition (SVD) in their paper [39], which included seven steps. The input image was first partitioned into overlapping blocks, after which each block was subjected to DCT, and the DCT coefficients were quantized. Following that, each quantized block was subdivided into non-overlapping sub-blocks. Each sub-block was subjected to SVD, after which features were retrieved to lower the dimension of each block using its largest singular value. All feature vectors were lexicographically sorted at the end, and duplicated picture image blocks were matched using a predetermined threshold. The experiment revealed that this algorithm could analyze and detect manipulation over images to which post-processing operations such as gaussian blurring, additive white gaussian noise, and JPEG compression were applied, in addition to detecting copy-move forgeries and locating duplicated regions. Park *et al.* presented a technique [40] that managed scaling, reflection, and rotation, among other geometric transformations. This technique analyzed likely reliable matched pairs based on the distance ratio between the most and second most similar match, using key points and descriptors from the image based on SIFT. The matched pairs were then sorted by their ratio value into a set of real matches.

Apart from the above-mentioned block-based and key point-based methods, many deep learning-based techniques have also come up to detect copy-move forgeries. Rao *et al.* proposed an automatic hierarchical feature representations learning model in their paper [68] to detect splicing and copy-move forgeries. They proposed an eight-layer CNN model with a fully connected layer and a two-way classifier. The kernel weights were set using the 30 basic high-pass filters in the first convolutional layer to improve generalization and speed up the network's convergence. RGB images were used to train the model. In their paper [41], Zhang *et al.* introduced CNN-based models for detecting copy-move forgeries. There were three convolutional layers with two max-pooling layers and two fully connected layers with a softmax layer in the fundamental models with two versions, *Siamese* (parameter sharing) and *pseudo-Siamese* (without parameter sharing). The model was fed a pair of images, one copied (C) and one original (O). Zhang *et al.* presented a two-stage deep learning methodology for detecting counterfeit photos [42]. The image was first converted to YCrCb space, after which it was segmented into 32 × 32 patches. To acquire the complex features, each patch was subjected to a three-level 2D Daubechies wavelet decomposition. Then, the patches were fed to the SAE model, which used a *multilayer perceptron (MLP)* layer to learn the complicated features. The contextual information from the patches was then combined after SAE processing to detect the image manipulation.

Wu *et al.* [43] proposed the BusterNet, a two-branch deep neural network-based CMFD model. Simi-Det and Mani-Det are the two branches that were used to detect cloned and modified regions, respectively. The CNN-based CMFD technique by Bayar *et al.* [44] used a confined convolutional layer to determine prediction error fields. They demonstrated the MISLnet CNN architecture, which included five convolutional layers for feature extraction with batch normalization and Tanh activation functions and three fully connected layers for classification with Tanh activation function and SoftMax layer. Table 1 represents a comparison amongst some of the above-mentioned deep learning-based copy-move image forgery detection techniques.

Table 1: Comparing deep learning-based CMFD techniques

Algorithm	*Approach & Features*	*Advantages*	*Limitations*	*Datasets Used*
Zhang et al. [41] (2016)	CNN based Siamese, pseudo-Siamese, 2-Chanel, and Hybrid 2-channel Siamese	Detection accuracy of 96.99%s	The small training set, large negative samples, and shallow network	INRIA Copydays, CoMoFoD, Image Manipulation Dataset, MICC-F2000, MICC-F220
Zhou et al. in [45] (2017)	rCNN, Tight blocking, and SVM classification	The accuracy rate is decent but has robustness against JPEG compression	Time-consuming and is not robust to many other post-processing techniques	Columbia gray DVMM, CASIA v1.0, CASIA v2.0
Rao et al. in [46] (2016)	Xavier-CNN, SRM-CNN, Patch wise sampling, SVM for classification	Detection Efficiency is decent	-	Columbia gray DVMM, CASIA v1.0, CASIA v2.0
Liu et al. [47] (2018)	SKPD, COB, CKN, K-nearest neighbor search, EM-based algorithm	Improved time complexity	Lacks robustness against many geometrical and post-processing operations	MICC-F220, CoMoFoD
Zhou et al. [48] (2016)	CPP network with a scalable color descriptor	High accuracy ratios with small false negative values show robustness against several post-processing operations like White Gaussian noise, blurring, JPEG compression, gamma correction	Not robust against geometrical operations such as translation, rotation, and scaling	CoMoFoD
Wu et al. [43] (2018)	VGG 16 architecture with percentile pooling and sigmoid activation function	78% opt-in sample accuracy with robustness to several attacks	Even though it shows robustness against several attacks, accuracy is quite low	CoMoFoD, CASIA v2.0
Zhang et al. [42] (2016)	SAE, 450-dimensional 3 Level 2D Daubechies wavelet Decomposition	91.09% detection accuracy along with forgery localization	Training time is high	Columbia gray DVMM, CASIA v1.0, CASIA v2.0

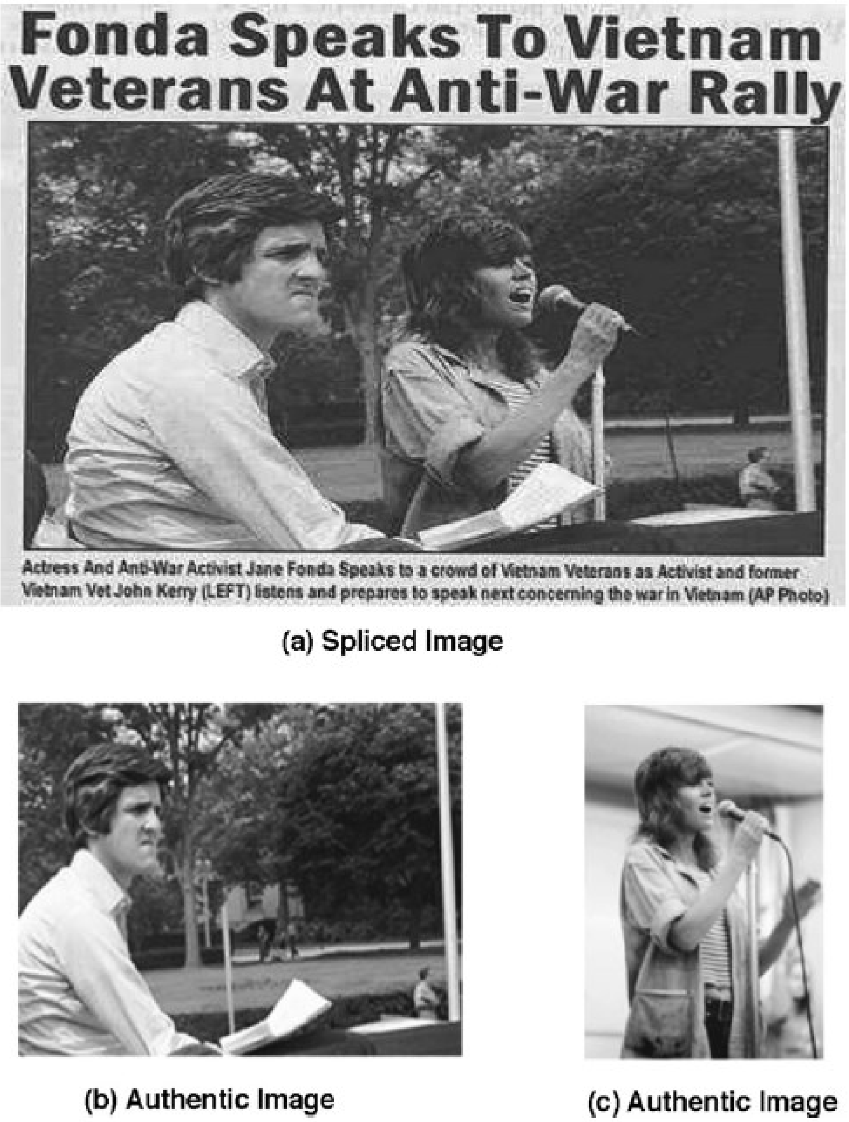
Apart from copy-move forgery, another highly used forgery technique is image splicing along with post-processing operations. The following section describes some of the benchmarked image splicing detection techniques presented by various authors to date.

2.3.3 Image Splicing Detection

Splicing is the process of replacing one or more elements of a host image with fragments from the other images, which could be used in malicious manipulation to create a scene that never existed to deceive the observers. Image splicing is a basic procedure of clipping and pasting sections from separate photos to create a new image without the need for post-processing, such as edge smoothing. Image splicing is one of the most basic and widely used image manipulation techniques. However, further image processing, such as smoothing the boundaries or removing the blur effect, can make it difficult to detect this manipulation. Image splicing is the basic technique of digital photomontage, in which photos are created by pasting images with the help of various editing tools such as Photoshop. It has become extremely popular, particularly to create "memes," as seen in a photo of Presidents Vladimir Putin and Donald Trump taken at the 2017 G-20 conference.

Just like the Catalan separatist demonstrations in October 2017, where some shocking photographs were deemed suspicious by the press [49], powerful images can be used to alter public opinion on a specific topic. During the 2004 presidential election campaign in the United States, a picture of Jane Fonda and John Kerry speaking together at an anti-Vietnam war demonstration was released and widely disseminated, as shown in Figure 8. It was eventually determined to be a spliced image made for political motives.

Some image statistics get disarranged when an image splicing operation is performed. On the other hand, the human visual system may not be able to detect these statistical changes. Even when an expert burglar performs postprocessing operations such as blending and matting on the forged image, the statistical disarrangements of the image cannot be attenuated. As splicing is frequently employed as the first step in image tampering, and splicing can be challenging to detect with modern image processing techniques, image splicing detection is critical in image tampering detection. For digital data forensics and information assurance, image splicing detection is very critical. People must be able to determine whether the given image has been spliced without any prior information. In another way, the splicing detection should be completely blind.



Fonda Speaks To Vietnam Veterans At Anti-War Rally

Actress And Anti-War Activist Jane Fonda Speaks to a crowd of Vietnam Veterans as Activist and former Vietnam Vet John Kerry (LEFT) listens and prepares to speak next concerning the war in Vietnam (AP Photo)

(a) Spliced Image

(b) Authentic Image

(c) Authentic Image

Figure 8: An image splicing example *[50]* (a) the spliced image of Jane Fonda and John Kerry, (b) and (c) authentic image of Kerry and Fonda, respectively.

Researchers have achieved significant progress in the field of image splicing detection technology. They have developed a variety of approaches, which are grouped into the following aspects as mentioned below [51]:

- Detection based on Noise patterns
- Detection based on Illumination conditions
- Detection based on the Image format

The first aspect takes advantage of the noise patterns, assuming that different images have varying noise patterns due to a mix of the camera makes/models, post-processing operations, and image capture conditions [52, 53, 54, 55, 56]. In general, human vision cannot discern an image that has been successfully spliced together. Still, the splicing procedure will change the image's statistical features, and the information about the changes can be employed for image splicing detection. Because the spliced portion came from a different image (the donor image) than the host image, the noise pattern in the spliced region may differ from the noise pattern in the rest of the image. As a result, the noise pattern may be used to detect the spliced region.

In prior studies, image analysis was performed using a high-order statistical property based on the wavelet transform [57, 58]. High-order statistical features can be used to represent the fundamental characteristics of natural images. Fu and Chen et al. proposed splicing detection based on the Hilbert-Huang transform [58, 59]. They applied the Hilbert-Huang transform on spliced images to generate features for classification considering the high non-linearity and non-stationary nature of the splicing operation. Similarly, Chen *et al.* [60] coupled the *Support Vector Machine (SVM)* classifier with the grey level co-occurrence matrix generated from the Block Discrete Cosine Transform (BDCT) domain to detect splicing. Sutthiwan et al. [61] integrated Markov features with edge statistical features in the chrominance domain for splicing detection. They extracted these image features from the Cr Channel, a chrominance channel in the YCbCr color space. These extracted image features were then fed into an SVM classifier to classify images as spliced or pristine.

The second aspect investigates the image's illumination conditions and uses *Color Filter Array (CFA)* interpolation patterns. Most digital cameras use a single image sensor with a CFA that produces one value per pixel to capture images. CFA interpolation (also known as demosaicing) is a technique for reassembling a full-color image by converting the captured output into three channels (RGB). Splicing can wreak havoc on CFA interpolation patterns in a variety of ways. For example, various cameras may utilize different CFA interpolation techniques, resulting in discontinuities when combining two images. Spliced regions are also frequently rescaled, which can cause CFA interpolation patterns to be disrupted. Because splicing is frequently done with two images, the lighting inconsistency of the spliced area can be used to detect spliced images. As a result, these artifacts can be used to help locate a spliced region. Kee *et al.* modeled the lighting environment and used the model's illumination environment consistency [61] to accomplish splicing detection.

The third aspect is based on the image format. JEPG images are the most used image storage format. As a result, the image-based JPEG format is used in several splicing detection systems. The third aspect takes advantage of the traces left behind by JPEG compression as it is a lossy compression format. These solutions rely on JPEG quantization errors or JPEG compression grid discontinuities [62, 63, 64, 65]. The original image is believed to have undergone sequential JPEG compressions in JPEG quantization-based approaches. However, the spliced section may have lost its initial JPEG compression features due to the smoothing or resampling of the spliced component. These characteristics can aid in the localization of a spliced regions. In addition, due to misalignment of the 8x8 block grids used in compression, spliced areas can be detected using JPEG grid-based algorithms. *JPEG Ghost* [65] and *Error Level Analysis (ELA)* are two methodologies that use JPEG compression traces. ELA can be used to detect areas in an image that have different compression rates. If a part of an image has a different error rate than the rest of the image, this might indicate that the image is digitally modified. In [66], Ramadhani used ELA along with *Laplacian Edge-Detector* to detect spliced images. He applies the ELA and Laplacian edge detection using the GIMP (GNU Image Manipulation Program) plugin. One of the significant issues with using ELA for image splicing detection is that it gives many false positives. This is often seen in cases where the JPEG image is of poor quality.

Table 2 below shows the number of research papers published on Image Splicing Detection based on different feature extractors between January 2010 and June 2021. Cells with higher intensity colors represent a more significant number of research papers. Most of these studies have been carried out between 2015 and 2021. The last five years have marked the rise of deep learning and computer vision, which is also depicted by the studies on image splicing detection based on deep learning models, as shown in Table 2.

Table 2: Number of studies on Image Splicing Detection between January 2010 to June 2021

Year	Markov Features	DWT	DCT	LBP	Deep Learning
2010	0	1	0	0	0
2011	0	1	0	0	0
2012	1	1	2	1	0
2013	0	0	1	1	0
2014	3	1	1	0	0
2015	2	3	2	3	0
2016	1	1	1	2	1
2017	1	1	1	0	0
2018	5	2	1	1	3
2019	1	1	1	1	6
2020	1	1	1	2	6
2021	0	0	1	3	2

Deep convolution neural networks (CNN) have recently demonstrated the ability to learn the image's deep-seated features. As a result, there has been a surge in interest in applying machine learning and deep learning algorithms to splicing detection and general image forensics. Classification approaches based on engineered or learned features have been developed by various researchers in the past couple of years. In each case, these strategies necessitate learning parameters or rules from a training set and applying these rules during the inference step. These methods detect or classify distinct types of manipulations using a learned classifier to detect them on a whole test image or on patches taken from a test image (although not all these methods explicitly target splicing attacks). Recently Patrick *et al.* [67] proposed an image splicing detection method in which they extract features using the Illumination-Reflectance model and Linear Binary Pattern (LBP). These features are then fed into a machine learning model such as SVM, *Logistic Regression, K-Nearest Neighbors*, etc., to get a computationally inexpensive solution.

Similarly, Jaiswal *et al.* [68] proposed a deep learning-based image splicing detection method in which they extract features using a pre-trained ResNet-50 model and feed these extracted features to three different classification models: *Naïve Bayes*, K-Nearest Neighbor, and SVM. Even though most researchers have lately focused only on image splicing detection, few studies have recently focused on the localization of the spliced regions in these spliced images. Rao *et al.* designed a CNN-based model that generates attention maps to represent the probability of splicing every pixel in the image [69]. In this method, convolution feature maps are fed to the attention module as inputs to generate the corresponding attention maps in which the pixel intensity represents the probability of being forged. Finally, Salloum *et al.* [70] presented an image splicing localizing method using a *Multi-task Fully Convolution Network (MFCN)*. The MFCN uses two output branches where one branch is utilized to learn the surface labels while the other branch is used to learn the edges of the spliced region. However, these approaches do not provide a satisfactory solution to the localization problem in spliced images. Thus, we propose an image forgery detection technique and investigate the forgery localization problem in forged images. Before discussing the proposed methods in detail, let us discuss the available benchmarked image forgery datasets and compare them.

2.4 Image Forgery Datasets

There exist several public datasets of forged images that can serve as benchmarks for algorithm evaluation. Table 3 presents a list of today's primary image forgery datasets and the characteristics of their content. Several factors are crucial when examining the usefulness of experimental datasets in evaluating forgery detection systems. The presence of ground truth binary masks for localizing the forged region is the first and foremost. We can only test forgery detection techniques without masks, and we cannot evaluate forgery localization strategies. Therefore, the dataset's scope is severely limited. In this regard, the CASIA v2.0 dataset, which is currently the largest realistic dataset accessible, has a severe flaw. Instead of ground truth masks, the dataset shows which two source photos were utilized to create each forged image. As a result, the only method to get solid ground-truth masks for the entire dataset is to use a semi-automated technique, which would be incredibly time-consuming considering the dataset's size. Pham *et al.* [71] have recently made ground truth masks for the CASIA v2.0 dataset available online, which helps overcome the flaw mentioned above.

Table 3: Benchmarked Image Forgery Datasets

Dataset Name	*Year*	*Image Formats*	*Coherent*	*Fake/Pristine Count*	*Total Size*
Columbia Monochrome [72]	2004	BMP grayscale	No	933/912	1845 images
Columbia Uncompressed [73]	2006	TIFF	No	183/180	363 images
First IFS-TC Image Forensics Challenge, Training [74]	2013	PNG (with possible JPEG history)	Yes	442/1050	1492 images
First IFS-TC Image Forensics Challenge, Phase 1 Testing [74]	2013	PNG (with possible JPEG history)	Yes	5713 unlabeled	5713 images
First IFS-TC Image Forensics Challenge, Phase 2 Testing [74]	2013	PNG (with possible JPEG history)	Yes	350/0	350 images
DSO-1 [75]	2013	PNG (with possible JPEG history)	Yes	100/100	200 images
DSI-1 [75]	2013	PNG (with possible JPEG history)	Yes	25/25	50 images
CASIA v1.0 [76]	2013	JPEG	Yes	921/800 (459 Copy-Move and 462 Spliced)	1721 images
CASIA v2.0 [76]	2013	JPEG, TIFF	Yes	5123/7491 (3295 Copy-Move and 1828 Spliced)	12614 images

The second characteristic is the image format of the dataset. Lossless formats, such as TIFF and PNG, have the advantage of allowing for uncompressed data, preserving the most sensitive traces required for noise-based and CFA-based approaches. On the other hand, JPEG-based techniques are unlikely to function on such datasets unless the images have a JPEG history and have been consecutively decompressed and encoded in a lossless format. Many JPEG-based algorithms may still operate in these circumstances. Despite the recent development of PNG files, JPEG remains the standard for Web-based forensics. In Table 3, only the CASIA v1.0 and CASIA v2.0 datasets [76] contain JPEG images, whereas most other datasets have PNG images with a possible JPEG history.

Finally, the third and last crucial characteristic of an image forgery dataset is the quality of the forged operation. Forgeries have been made artificially in some datasets by automatically changing certain sections or putting a section of one image into another. Amongst the datasets provided in Table 3, the two Columbia datasets [72, 73] fall into this category. Thus, after comparing all these datasets, CASIA v2.0 was selected, considering the number of authentic and forged images available and the newly available ground-truth masks.

2.5 Class Imbalance

Many real-world applications naturally inherit high-class imbalance, and thus efficient classification of such imbalanced datasets is a vital area of research. Furthermore, excessively imbalanced data makes classification and segmentation even more difficult as many deep learning models will get biased towards the majority group. In extreme circumstances, these deep learning models might even disregard the minority group entirely [77]. As mentioned by Johnson *et al.* [77], research in this area is minimal. Class imbalance occurs when the minority class contains considerably fewer samples than the majority class in a binary classification problem that uses data from two classes. The minority group, i.e., the positive class, is the class of interest in many problems like fraud detection, forgery detection medical imaging lesion detection [77, 78]. The effect of class imbalance in image segmentation differs from image recognition as the background class is the majority class in image segmentation, which has diverse characteristics and highly robust segmentation accuracy. Figure 9 shows the ground-truth mask of various tampered images in the CASIA v2.0 dataset. It can be seen that the foreground pixels(white) are pretty less compared to the background pixels (black) that depict the class imbalance in the dataset. It can cause the model to overclassify the background class because of its higher prior probability.

Furthermore, because foreground classes are primarily used to evaluate segmentation performance, the focus is on improving accuracy in those classes. When there is a class imbalance in training data, learners are more likely to overclassify the majority class because of its higher prior probability. As a result, the pixels belonging to the minority class are more frequently misclassified than pixels from the majority class. Due to these adverse effects, achieving the goal of effectively predicting the positive class of interest is quite difficult.

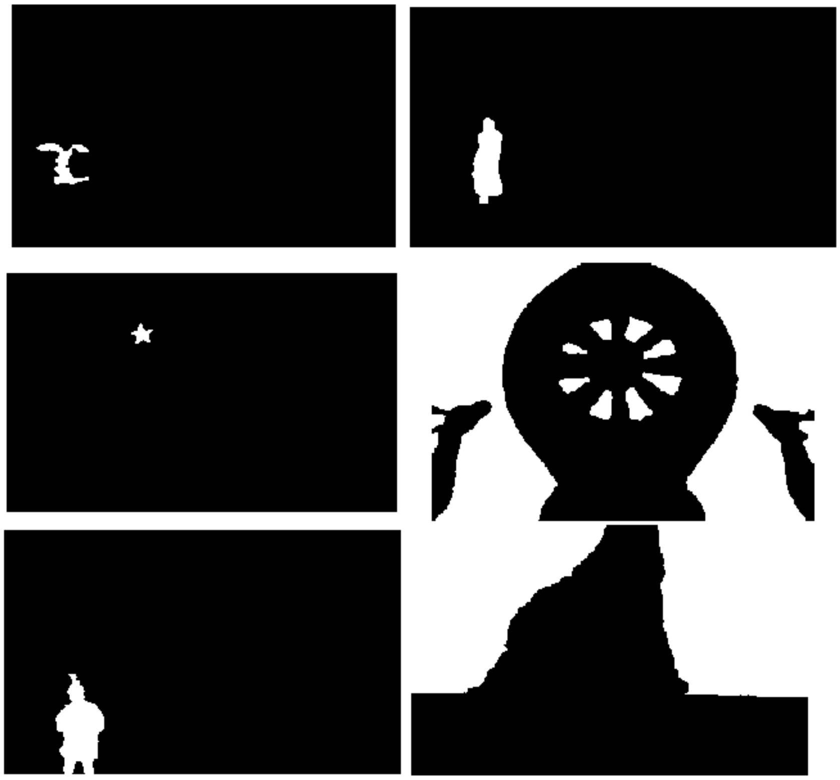


Figure 9: Ground-truth masks of forged images in CASIA v2.0

Deep learning approaches have become popular over the previous ten years as they have enhanced state-of-the-art image recognition, speech recognition, and various other fields [79].

Increased data availability, different algorithmic innovations that speed up training and improve generalization to new unseen data, advancements in hardware and software are all factors in their recent success [80]. Despite these advancements, very little statistical work has adequately examined strategies for dealing with class imbalance using deep learning. Many researchers agree that deep learning with imbalanced data is an understudied topic [77, 81].

When learning from unbalanced data, it is critical to examine the representation of the minority and majority classes. By developing artificial data sets with varied combinations of complexity, degrees of imbalance, and training set size, Japkowicz [82] investigated the impacts of class imbalance. The results showed that as the problem complexity increases, the sensitivity to class imbalance also increases.

It is possible to reduce the bias towards the majority class by changing the model's underlying learning or decision mechanism to increase sensitivity to the minority class or by changing the training data to reduce imbalance [77]. To address the class imbalance, conventional approaches like re-weighting and resampling can be used to degrade the majority class's accuracy [83]. These methods can be divided into algorithm-level techniques and data-level techniques. The learning or decision-making process is altered in algorithm-level techniques to give the positive class (minority class) more importance. The decision threshold is typically lowered to reduce bias towards the negative class, or algorithms are tweaked to account for class weights. These techniques include using various loss functions such as weighted cross-entropy loss and dice loss, which will be discussed in the next section. Data-level methods for tackling class imbalance include under-sampling and over-sampling [77]. These methods change the training data to reduce the level of class imbalance. Under-sampling discards data voluntarily, lowering the overall amount of data from which the model will learn. As a result, the model may miss out on learning essential information that could have been learned from the samples that were deleted due to under-sampling [84]. Oversampling involves randomly selecting samples of the minority class and duplicating them to increase the minority class samples in the dataset [85]. However, a single sample might get selected multiple times to get resampled. As a result, the oversampled dataset can have multiple copies of the same sample belonging to the minority class. Due to the larger size of the oversampled training set with multiple copies of a given sample, over-sampling will increase training time and produce over-fitting [86]. Thus, algorithm-level techniques can be more helpful at tackling class imbalance in semantic segmentation. The following section discusses in detail some of the algorithm-level techniques used in this thesis.

2.6 Loss Functions

Most of the time, datasets will have some level of class imbalance. This problem of imbalanced datasets is often seen in image segmentation tasks. This section discusses in detail various loss functions that can be used to tackle the class imbalance problem for image segmentation.

As seen in Section 2.5, the images in CASIA v2.0 suffer a high-class imbalance. It means that the background class (black) is the majority class and the foreground class (white, forged region) is the minority class, as shown in Figure 9. Many data-level techniques such as under-sampling and oversampling can be used to balance the dataset. However, as discussed in Section 2.5, both oversampling and undersampling can lead to new issues such as overfitting and loss of important information, respectively, and do not directly tackle the issues caused by class imbalance data. To prevent this, an algorithm-level technique such as a weighted loss function can be used. A weighted loss function considers class weights that are inversely proportional to the class frequencies in the image, i.e., the minority class will have a higher-class weight, and the majority class will have a lower-class weight. It gives more importance to the loss of the minority class and makes the model focus more on the minority class than the majority class.

$$WBCE = -\frac{1}{N}\sum_{i=1}^{N} w_1 y_i \cdot log(p(y_i)) + w_0(1-y_i) \cdot log(1-p(y_i)) \tag{1}$$

Equation 1 depicts the weighted binary cross-entropy loss function where $\mathrm{y_i}$ is the positive class (forged region) and $\mathrm{p(y_i)}$ denotes the probability function. In the above equation w_0 represents the class weight for the majority class whereas w_1 represents the class weight for the minority class. The minority class, also called the positive class, is denoted by $\mathrm{y_i}$ in the above equation. A higher value of w_1 can be used so that when a pixel belonging to the minority class is classified incorrectly, the prediction error becomes larger, thus making the model pay more attention to the minority class [87, 88]. Some image segmentation metrics can also be used as loss functions for the task of image segmentation. One such highly known metric is the Dice Coefficient (DSC). When it comes to class imbalance, the dice coefficient can perform well as a loss function, unlike other extensively used loss functions like Binary Cross-Entropy (BCE) loss function [89]. In contrast to BCE, the dice coefficient just considers the segmentation class and ignores the background class. In image segmentation, pixels can be classified as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). These measures are shown in Equations 12, 13, 14, and 15 in Section 4.3. The dice coefficient can be represented by using these measures, as shown in the following equation.

$$DSC = \frac{2TP}{2TP + FN \ + FP} \tag{2}$$

$$Dice\ Loss = 1 - DSC \tag{3}$$

As it does not consider the true negatives, it does not dominate over the minority class. It produces a score in the range of [0,1], where 0 represents no overlap and 1 represents perfect overlap. Therefore, by subtracting the dice coefficient from 1, we get the dice loss as shown above [90]. The losses mentioned above have been used to analyze and investigate the localization problem of forged regions in the images of CASIA v2.0 in Section 4.3. The next chapter will define the proposed methods used in this thesis.

Chapter 3

The Proposed Method

In this section, we present an image forgery detection method along with quantitative results utilizing various metrics. We also study the forgery localization problems using various deep learning segmentation models. The following sections first give an overview of our proposed model and then explain all the elements of its architecture.

3.1 Overview

The availability of low-cost, high-resolution digital cameras and the rapid expansion of user-friendly and complex digital image editing programs have increased the difficulty of assuring digital image authenticity. It makes distinguishing authentic and tampered sections quite tricky when trying to locate forged sections. Thus, for combating malicious forgery, detecting and localizing image forgery have become critical. As a result, the development of reliable imagery authenticity verification tools is vital in today's digital world. Thus, this thesis presents a passive method for detecting image forgery based on texture categorization.

The proposed method deals with detecting forged images by classifying the images as forged or pristine. It is named as Image Forgery Detection Network (IFD-Net) and consists of the following steps:

- Converting the RGB channels of the image to YCbCr channels and extracting the Cr channel,
- Calculating the local binary pattern (LBP) and applying it to the Cr channel,
- Finally, feeding these images to a ResNet-50 model for training the IFD-Net v1.

Figure 10 provides a pictorial representation of the training phase pipeline of IFD-Net v1 for classifying the images as forged or pristine. At every step, we can see what type of preprocessing is applied to the image before it is fed to the *ResNet-50* model. Figure 11 shows the pipeline for the testing phase of IFD-Net v1, which is used to evaluate the trained model on the test data.

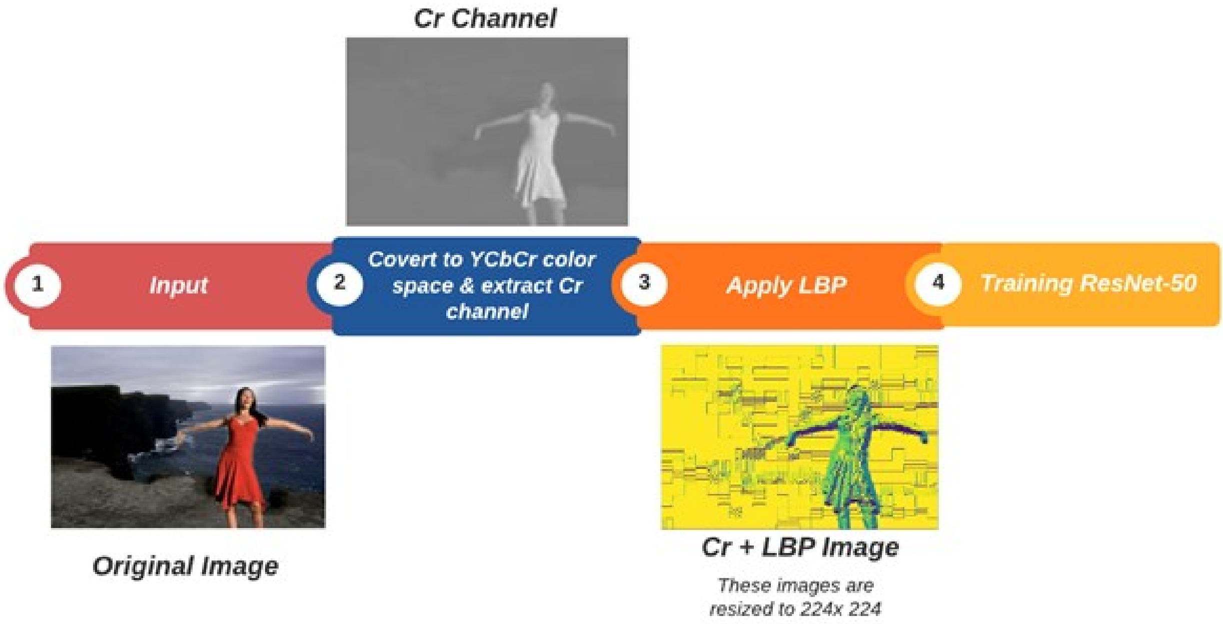

Cr Channel
1
Input
2
Covert to YCbCr color space & extract Cr channel
3
Apply LBP
4
Training ResNet-50
Original Image
Cr + LBP Image
These images are resized to 224x 224

Figure 10: IFD-Net v1: Classification of Images as forged or pristine, training phase pipeline

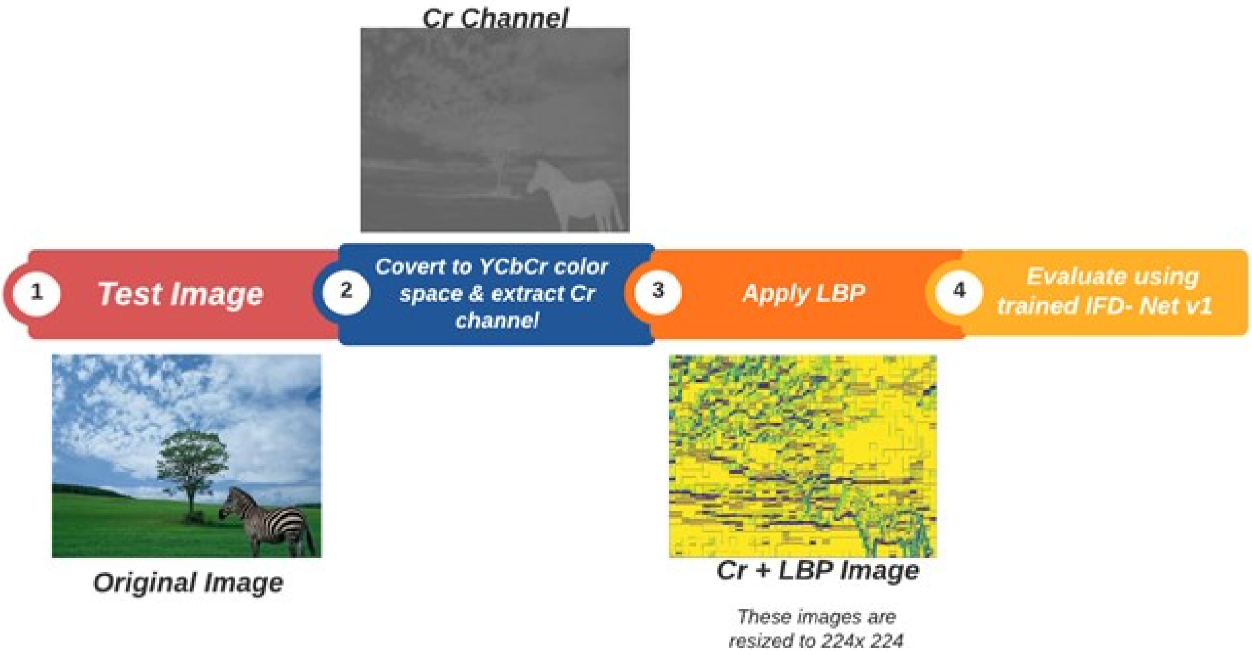

Cr Channel
1
Test Image
2
Covert to YCbCr color space & extract Cr channel
3
Apply LBP
4
Evaluate using trained IFD- Net v1
Original Image
Cr + LBP Image
These images are resized to 224x 224

Figure 11: Testing Phase Pipeline for IFD-Net v1

Like the IFD-Net v1, this thesis also presents a version two known as the IFD-Net v2. IFD-Net v2 consists of the following steps:

- Extracting the noise residual map using the Noiseprint Model,
- Applying *Adaptive Histogram Equalization (AHE)*,
- Finally, feeding these images to the ResNet-50 (pre-trained) model for training the IFD-Net v2.

Figure 12 provides a pictorial representation of the training phase pipeline of IFD-Net v2 for detecting forged images. Each step shows how the original image is modified using preprocessing steps such as extracting noise residual maps and applying AHE before it is fed to the ResNet-50 model. Figure 13 shows the pipeline for the testing phase of IFD-Net v2. Similar preprocessing steps as the training pipeline are followed, and then these preprocessed images are used to evaluate the trained model's performance.

As detecting images as forged or pristine is a binary classification task, both IFD-Net v1 and IFD-Net v2 use the Binary Cross-Entropy loss as shown in Equation 4 where $\mathrm{y_i}$ is the positive class (forged images) and $\mathrm{p(y_i)}$ denotes the probability function.

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i)) \tag{4}$$

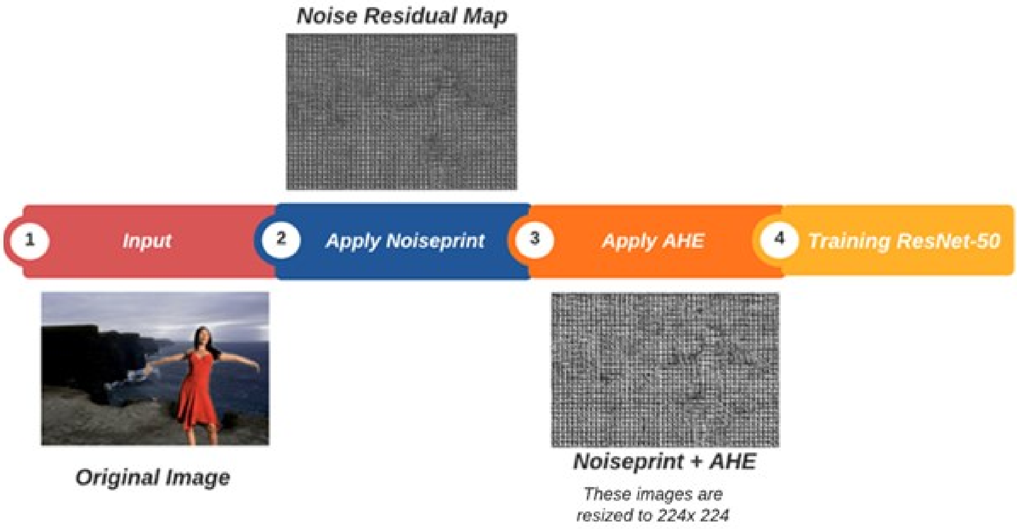


Figure 12: IFD-Net v2: Classification of Images as forged or pristine, training phase pipeline

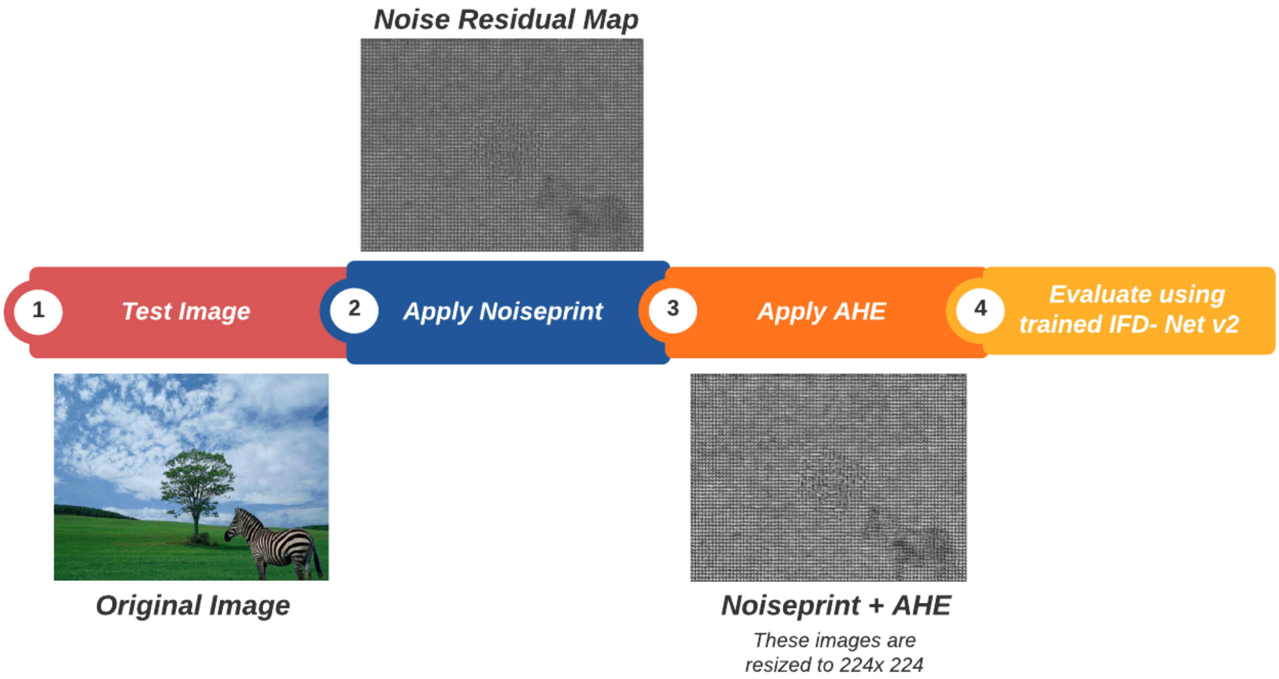


Figure 13: Testing Phase Pipeline for IFD-Net v2

To study the forgery localization problem, an Image Forgery Localization Network (IFL-Net) is used. It consists of the following steps:

- The first two steps are feeding the forged image (CR+LBP image) and its corresponding ground-truth mask to step three.
- In step three, these images and ground-truth masks are then fed to a modified U-Net model for training the IFL-Net for the task of *semantic segmentation.*

Figure 14 provides a pictorial representation of the training phase pipeline of IFL-Net for analyzing the localization of the forged regions in the images present in the CASIA v2.0 dataset. As discussed in Section 2.6, Weighted Binary Cross-Entropy (WBCE) loss and the Dice Loss are used for training the IFL-Net in accordance with the highly imbalanced data in the CASIA v2.0 dataset. Figure 15 provides a pictorial representation of the testing phase pipeline of IFL-Net, which is used to evaluate the trained model on the test data.

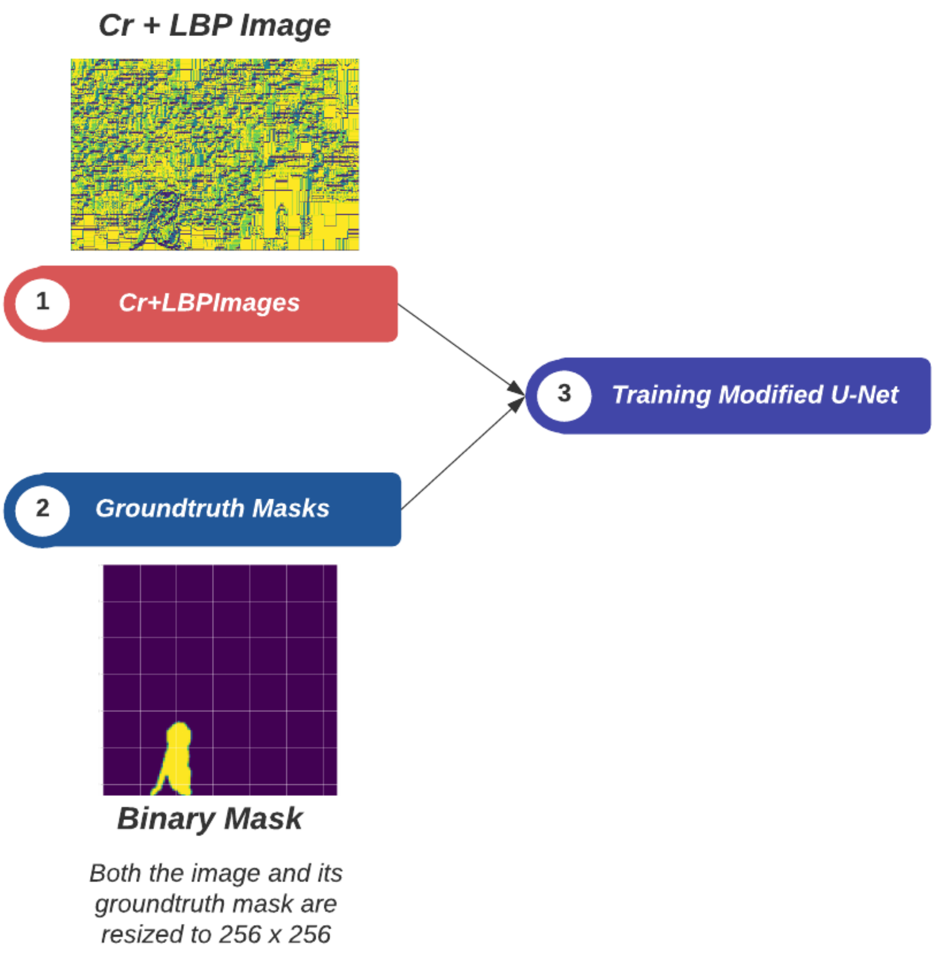


Figure 14: IFL-Net: Localization of Forged Regions, training phase pipeline

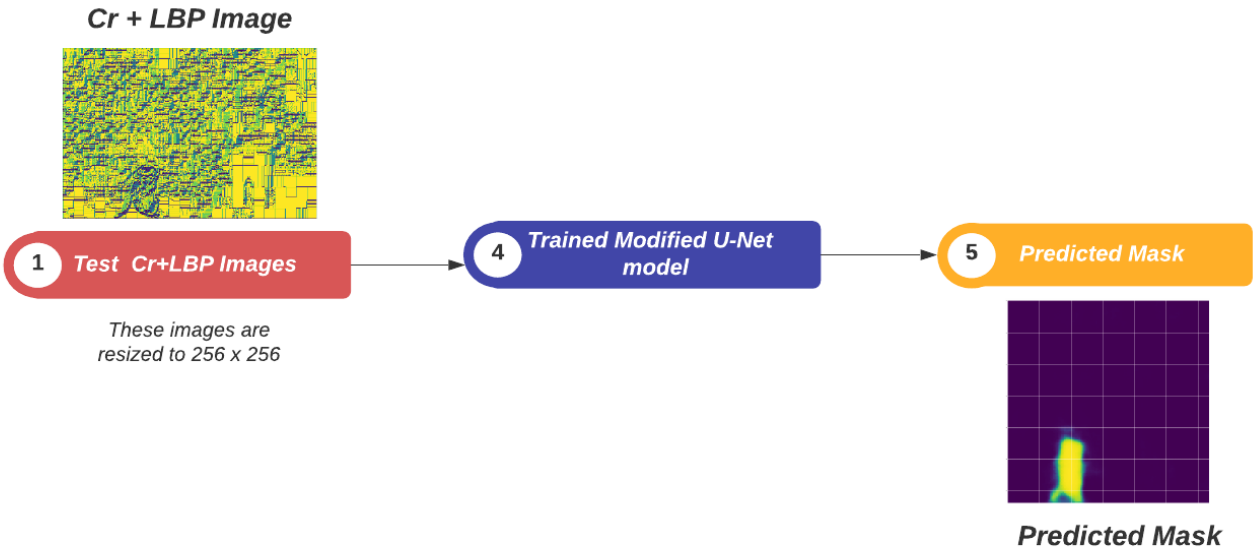

Cr + LBP Image
1
Test Cr+LBP Images
These images are resized to 256 x 256
4
Trained Modified U-Net model
5
Predicted Mask
Predicted Mask

Figure 15: Testing Phase Pipeline for IFL-Net

The following sections explain the above-mentioned image forgery detection methods and investigates the localization problem in detail.

3.2 Extracting the Red-Difference Chroma Component

RGB color space displays colors by combining the red, green, and blue components of the color. Any color can be displayed using these three components. YCbCr is a color space family used in video and digital photography systems as part of the color image pipeline. The luminance component is Y, whereas the blue-difference and red-difference chroma components are Cb and Cr. The primary goal of this step is to concentrate on the characteristics of a single channel. Usually, RGB color spaces are used by image forgers to modify images and wrap modified traces. The chrominance spaces, which are regarded as an efficient means of identifying forged images, are utilized in this thesis to detect and localize forgery in digital images. The RGB input is converted to YCbCr representations using the equations (5), (6), (7), and (8), as shown below.

$$Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \tag{5}$$

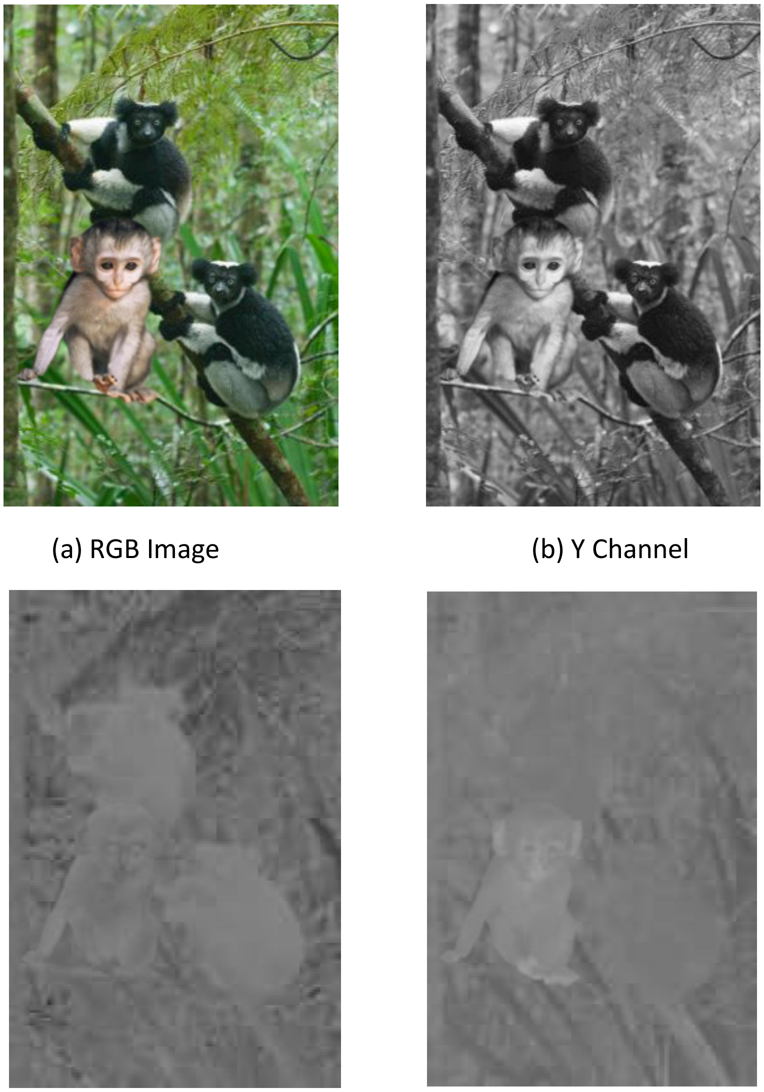
$$Cr \leftarrow (R - Y) \cdot 0.713 + delta \tag{6}$$

$$Cb \leftarrow (B - Y) \cdot 0.564 + delta \tag{7}$$

Where,

$$delta = \begin{cases} 128, & for\ 8 - bit\ images \\ 32768, & for\ 16 - bit\ images \\ 0.5, & for\ floating - point\ images \end{cases} \tag{8}$$

The chroma spaces are created by subtracting luminance from red (Cr = R-Y) and blue (Cb = B-Y). The color is stored in the YCbCr color space in terms of luminance and chrominance, with chrominance being less sensitive to human eyes than luminance. The edge sensitivity and sharpness of the forged objects are more noticeable in the chroma channels of an image than in the RGB channels. Even though the altered image appears natural, there is some evidence of tampering in the chrominance channels. Generally, humans are more sensitive to luminance than to chrominance in an image. Even though a forged image may look reasonably natural to the human eyes, some traces of forgery are always left in the chrominance channels. Forged edges in Cb or Cr components are not as smooth as the original RGB image edges. Thus, in this thesis, images are converted to chroma spaces to take advantage of such traces of tampering caused by the forgery operation.



(a) RGB Image

(b) Y Channel

(c) Cb Channel

(d) Cr Channel

Figure 16: (a) is the original RGB forged image, and (b), (c), and (d) are the corresponding Y, Cb, Cr channels, respectively.

A color image with its luminance and chrominance components is shown in Figure 16. It can be observed in these images that the monkey’s contours (which represent the forged region) are covered up and smooth in the Y component compared to the Cb and Cr components. The contours of the monkey are sharper than the rest of the objects in the Cr channel image. Thus, the edges of the forged region will be more detectable in the Cr channel.

Following the extraction of the chroma component of the image, to further extract the image's textural features, a texture categorization feature is applied to the chroma component, as discussed in the following section.

3.3 Local Binary Pattern (LBP)

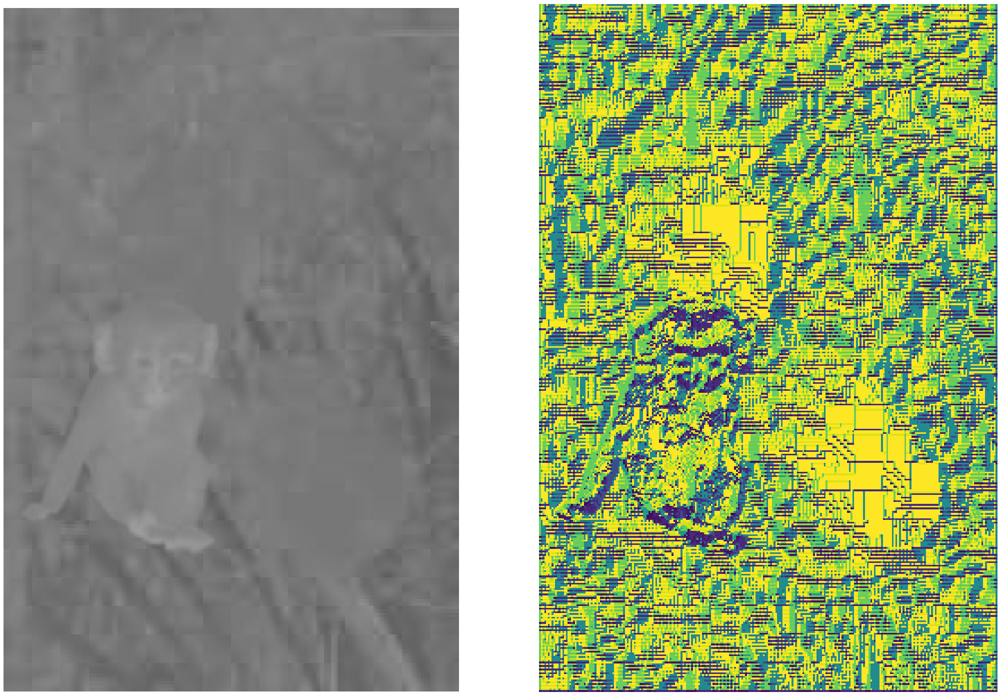
Local Binary Pattern (LBP) is a powerful texture categorization feature. LBP is a useful texture description operator for images used to classify image content and texture description. The spatial arrangement of color or intensity in an image or a selected part of an image is described by image texture. Face recognition is one of the most well-known applications where LBP features are used. In recent years, image mosaic detection has increasingly used a combination of LBP and conventional methods.

In a rectangular window, a simple LBP operator is calculated. In a circular neighborhood $\mathbf{(p, r)}$, LBP is computed using $\boldsymbol{p_c}$ as the central pixel value, $\mathbf{p}$ as the number of neighborhood pixels, and $\mathbf{r}$ as the radius of the neighborhood. Equations 9 and 10 depict the LBP operator where p_i represents the pixel value of the i_{th} pixel.

$$LBP_{p,r} = \sum_{i=1}^{p-1} S(p_i - p_c) \cdot 2^i \tag{9}$$

$$S(p_i - p_c) = \begin{cases} 1, & p_i \geq p_c \\ 0, & p_i < p_c \end{cases} \tag{10}$$

The results of this thesis are presented using circular LBP. The rationale for adopting LBP is that when cut and paste operations are applied to an image, the texture consistency in the image is broken. This textural inconsistency can be captured using LBP, which highlights such micro-patterns in an image. The main advantages of LBP over other descriptors are that it is invariant to monotonic illumination changes, it is rotation invariant, and it has a low processing complexity. It assigns a binary number to each pixel in the image by thresholding the neighboring pixels with the center pixel. The LBP operator is applied to each image to highlight the forgery artifacts, i.e., the sharp edges along the boundary of the forged region and the micro-edges inside the forged region. This makes the forgery artifacts more prominent because of LBP's capacity to capture micro-patterns in an image [91, 92]. Figure 17 shows the texture pattern after applying LBP to the Cr channel image extracted above. It can be observed that the texture pattern at the edges of the forged region is different when compared to the rest of the texture pattern of the image. It forms a kind of boundary around the forged area by capturing the micro-patterns. Thus, LBP is appropriate for emphasizing the forgery artifacts and making them more prominent in the tampered image [91]. In addition to the textural features, the noise inconsistencies introduced in the image after the forgery operation can also be used to detect forged images.



(a) Cr Channel Image

(b) LBP Texture Pattern

Figure 17: LBP Texture Pattern of Cr Channel Image

The following section explains how these noise inconsistencies can be used to generate a noise residual map that can help detect forged images. These noise residual maps are used in the IFD-Net v2 for detecting forged images.

3.4 Obtaining noise residual map using the Noiseprint

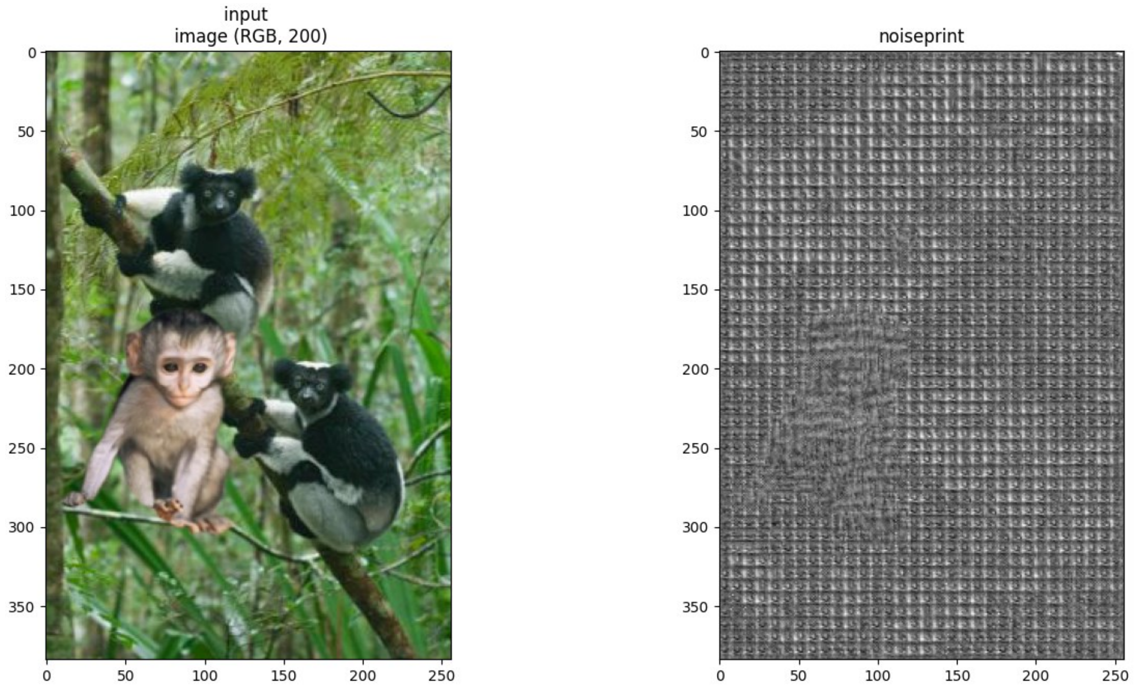
When the high-level semantic (scene) content is subtracted or removed from an image, the remaining noise-like signal is known as the *Noise Residual* of the image. Such noise residuals can be extracted from images using high-pass filters in the spatial or transform domain (DCT, DWT) or by denoising algorithms [93]. *Photo-response non-uniformity (PRNU)* is one of the most effective ways for extracting noise residuals from digital photos. However, there are two significant drawbacks to this method: first, it requires multiple photos to analyze the camera fingerprint, and second, its effectiveness is hampered by the low power of the signal of interest in comparison to noise. Cozzolino et al. [93] presented Noiseprint, a new approach for extracting the noise residual from images to tackle these two drawbacks. This technique focuses on camera-specific noise-based fingerprints while suppressing image content. Compared to other top-performing approaches, including a PRNU-based method, the Noiseprint model outperforms them [94].

The camera-specific fingerprints can be extracted using a variety of approaches. The Noiseprint model, on the other hand, is the most recent advancement. The Noiseprint model is a CNN-based Siamese network that outputs a noise residual map that consists of traces of the camera model artifacts rather than the traces of imperfections of individual devices. The noise residual map that the Noiseprint model generates is an image size pattern in which camera model-related artifacts are emphasized by removing the high-level scene content in an image. In most cases, spliced images are made up of photographs obtained by various camera models. During the image capture process, each camera model creates its unique fingerprint. As a result, these camera-specific fingerprints may be retrieved and used to uncover the picture splicing manipulation by using Noiseprint. However, in copy-move forged images, these camera-specific fingerprints remain the same throughout the image as the copy-move forged region belongs to the same image. Thus, the noise residual map that Noiseprint outputs for a copy-move forged image might be like the noise residual map of its original image as both have the same camera fingerprint. This makes it difficult to distinguish between pristine and copy-move forged images based on their noise residual maps. The Noiseprint model is used as a preprocessing step in the IFD-Net v2 to obtain the noise residual from the images so that the IFD-Net can also be evaluated by taking advantage of the noise inconsistencies introduced due to the forgery operation.

Additionally, a contrast boosting technique known as Adaptive Histogram Equalization (AHE) is applied to all the noise residual maps. AHE is an image processing approach to enhance image contrast that can enhance the noise residual map's contrast. The adaptive equalization approach differs from traditional histogram equalization to compute multiple histograms, each corresponding to a different portion of the image and uses them to redistribute the image's brightness values. As a result, it is ideal for boosting local contrast and sharpening edge definitions in different parts of an image. Thus, AHE is applied to the noise residual maps generated using the Noiseprint model. Figure 18 shows an example of a spliced image and its corresponding noise residual map using the Noiseprint model to which AHE has been applied. It can be observed from the noise residual map of the spliced image that the spliced region (monkey) has a different noise residual pattern when compared with the rest of the image. This can be quite useful to distinguish spliced images from pristine images.

3.5 Image Augmentations

Many Computer Vision tasks have shown that deep *convolutional neural networks* perform exceptionally well. However, to avoid overfitting, these networks rely extensively on huge data. In many cases, imbalanced classes can be a further stumbling block. While there may be enough data for certain classes, under-sampled classes will suffer from poor class-specific accuracy. Overfitting occurs when a network learns a function with extremely high variance to model the training data perfectly. When a model is trained on only a few examples of a class, it is likely to overfit the small training dataset, causing the trained model to perform poorly on test data. Unfortunately, many application fields, such as medical image analysis and image forgery detection, do not have access to huge data. There are various approaches to deal with these problems that come with limited data in deep learning.

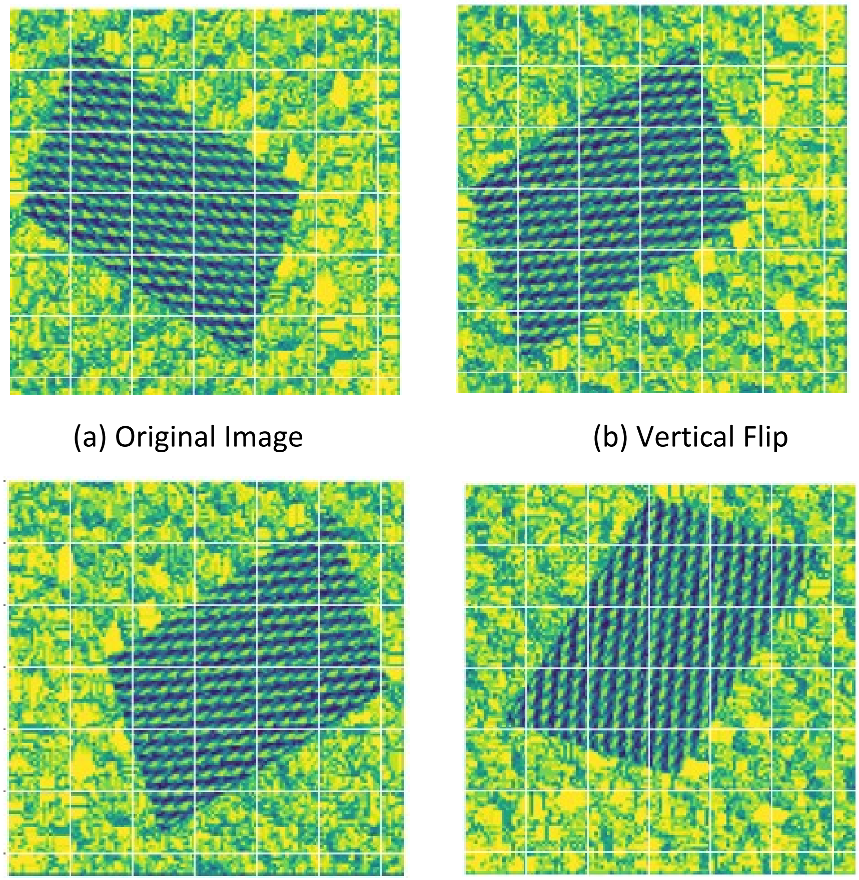


(a) Spliced Image

(b) Noise Residual Map + AHE

Figure 18: Noiseprint and AHE applied to a Spliced Image

Data Augmentation is a term that refers to a range of strategies for increasing the size of training datasets so that deep learning models can be made more efficient. Image augmentation is a handy strategy for increasing the size of the training set without having to acquire new photos while creating convolutional neural networks. The concept is straightforward: replicate images with slight variations so that the model can learn from additional examples. These enhancements can be combined to create a variety of variations of the original image. However, augmentation will be detrimental if it produces images that are considerably different from those used to evaluate the model; thus, it must be done carefully.



(a) Original Image

(b) Vertical Flip

(c) Horizontal Flip

(d) Random Right Angle Rotation

Figure 19: Image Augmentations

The question that arises is, “What is the impact of augmentation on prediction accuracy?” Extending the training dataset with different augmentation approaches and seeing which improves performance the most can help answer the above question. However, it is time-consuming and must be done with care to ensure that the augmented images do not differ much from the images used to evaluate the model as it can have a negative impact. Augmentations must only be applied to the training set. Validation and test sets aim to evaluate the model's performance in a realistic application. Therefore, duplicated images may artificially enhance performance measures.

Several geometric augmentation strategies include vertical and horizontal flipping, and random right-angled rotations. Figure 19 shows an example of these augmentations applied to an LBP image. Vertical and horizontal flipping are simple image augmentation techniques that flip the input image either vertically or horizontally. *Right-angled rotation* is an image augmentation strategy that rotates the image by 90 degrees. To analyze the impact of these geometric augmentations on the detection accuracy of IFD-Net, we carry out experiments with augmented data and the original data. Section 4.2 discusses the impact of using these geometric augmentations on the prediction accuracy of the IFD-Net. Before discussing the impact of these geometric augmentations, let us first go through the architectures of both IFD-Net and the IFL-Net.

3.6 Classification Using ResNet-50

In discriminative tasks, deep learning models have made amazing progress. Deep network architectures, sophisticated processing, and access to massive data have all contributed to this. Thanks to the convolutional neural network discovery, deep neural networks have been effectively applied to Computer Vision applications such as object detection, image classification, and image segmentation. The spatial properties of images are preserved using parameterized, sparsely connected kernels in these neural networks. Convolutional layers down sample images' spatial resolution while growing the depth of their feature maps progressively.

Deep neural networks with additional layers allow a deep neural network model to contain more parameters, which increases the degree of freedom of the model. The ability to learn additional complex features will increase as the model becomes more complex. When a neural network is given complete freedom to choose parameters without regularization, the likelihood of a global minimum is reduced, and the model instead finds local minima. As a result, regularization approaches effectively regulate the most in-depth neural networks and attempt to avoid model over-fitting behavior. However, even after using regularization methods, the model can still overfit. To avoid such behavior while maintaining the benefits of deep neural networks, researchers have devised a novel architecture known as Residual Network [95].

The vanishing gradients problem, in which gradients at the last layer cannot propagate back to the initial layers, is one of the most severe issues with deep neural networks. As a result, learning will be protracted and ineffective. Shortcut connections in Residual blocks (identity block shown above in Figure 20) allow a model to learn the input’s identity mapping quickly. The shortcut connections also aid in carrying gradients back to starting layers without the issue of vanishing gradients. The problem of class imbalanced data in real-world applications is quite prominent and can make neural networks biased towards the majority class in the dataset. Ding *et al.* [96] performed various experiments on very deep CNN architectures (like 50 layers) to determine their performance on imbalanced datasets. They observed that as deeper neural networks have more local minimums with acceptable performance, it becomes easier for gradient descent to find acceptable solutions. Thus, we use the ResNet-50 architecture (pre-trained) rather than a simpler network such as AlexNet [97] for the binary classification of images as forged or pristine. Figure 32 in the Appendix shows the overfitting behavior of AlexNet at the task of binary classification of the images in the CASIA v2.0 dataset.

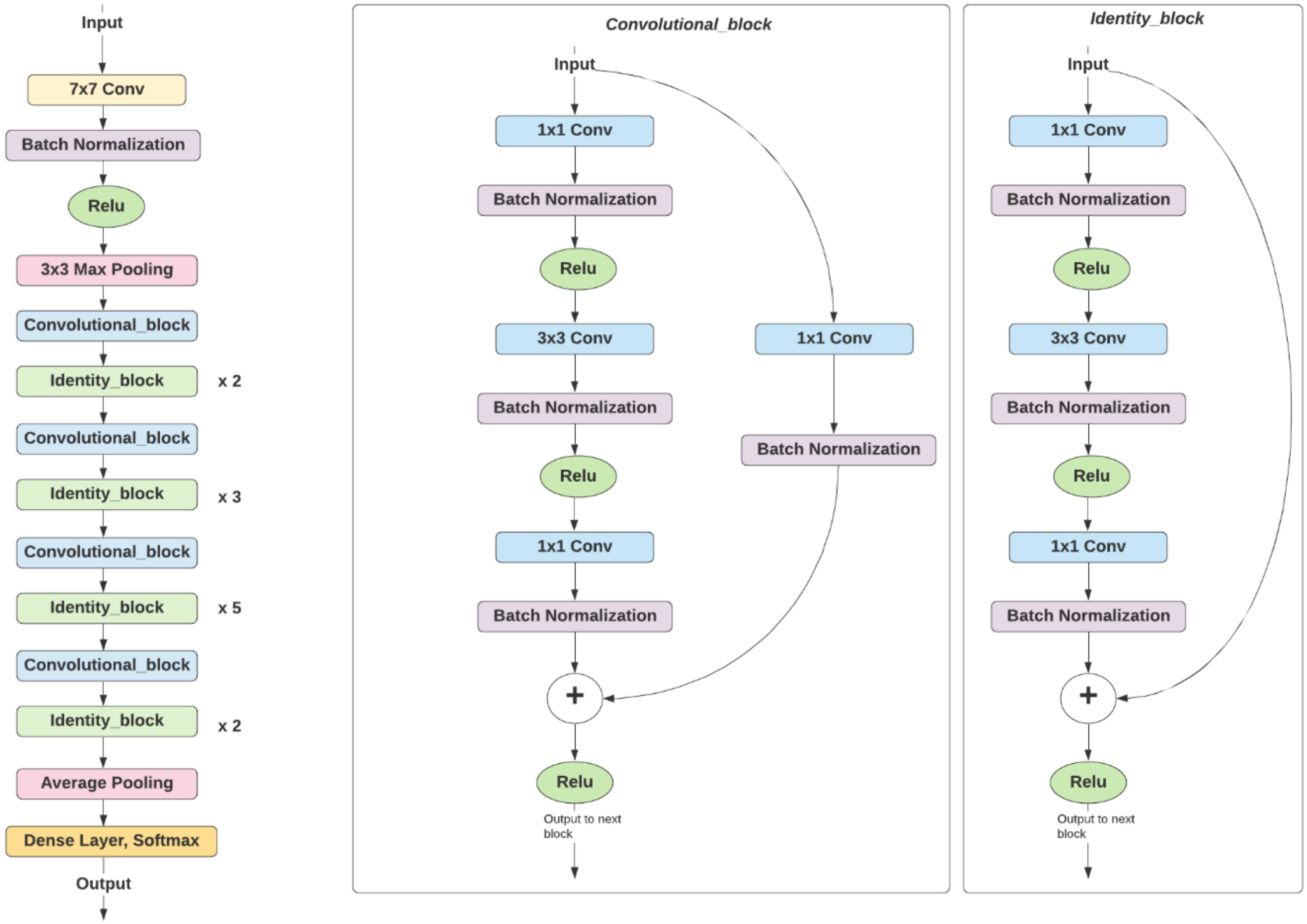


Figure 20: IFD-Net Architecture (ResNet-50)

The ResNet-50 model consists of a total of 50 layers. These layers are grouped together in residual blocks of 3 layers, as shown in Figure 20. There are two types of residual blocks: the convolutional block and the identity block. Both the blocks consist of a 1x1 convolutional layer, a 3x3 convolutional layer, and finally, a 1x1 convolutional layer at the end. The output of the first two layers undergoes batch normalization followed by the RELU activation function. However, the output of the 3rd layer is first batch normalized, then added together with the shortcut connection, and then goes through the RELU activation function as shown in Figure 20. In the identity block, the input to the block is added to the output of the last layer using the shortcut connection. However, unlike the identity block, the shortcut connection consists of a 1x1 convolution layer in the convolutional block. This 1x1 convolutional layer in the shortcut connection helps adjust the number of channels and the resolution before the adding operation. Multiple convolutional and identity blocks are used in conjunction with each other to form the ResNet-50 architecture, as shown in Figure 20. The final output layer is a Dense layer with a Softmax activation function that outputs the class probabilities. This architecture is in accordance with the ResNet-50 architecture as proposed by He *et al.* [95].

The IFD-Net v1 and v2 have the same network architecture as shown above in Figure 20. However, the input differs in both cases. The IFD-Net v1 uses Cr + LBP images as inputs to the ResNet-50 model. Similarly, the IFD-Net v2 uses Noiseprint + AHE images as inputs to the ResNet-50 model. Then these inputs are used to train the ResNet-50 model and classify the test images as forged or pristine.

In addition to forgery detection, we also analyze the problem of forgery localization using the CASIA v2.0 dataset. The following section describes the various segmentation models used to analyze the forgery localization problem.

3.7 Segmentation model for Analyzing Forgery Localization

This thesis presents a convolutional neural network-based image forgery localization methodology using *Image Segmentation*. Image segmentation refers to partitioning the image into different segments such that each segment represents a different entity. The IFL-Net uses a modified version of *U-Net*, a convolution neural network developed for the task of biomedical image segmentation [98]. In contrast to classification, semantic segmentation necessitates pixel-level discrimination and a technique to project the discriminative features learned at various stages of the encoder onto the pixel space.

As discussed in Section 2.5, class imbalanced data makes it difficult for neural networks to generalize well as they tend to get more biased towards the majority class. However, Ding *et al.* [96] show the error surfaces of deep neural networks exhibit better training convergence properties than shallower neural networks. They show that as deeper neural networks have more local minimums with acceptable performance, it becomes easier for gradient descent to find acceptable solutions. Thus, to apply this to semantic segmentation for localizing the forged regions in forged images, we propose a modified U-Net architecture that uses a deep ResNet-34 model as the encoder of the modified U-Net. Additionally, ResNet-34 is used as the encoder of the U-Net model because of its ability to tackle the vanishing gradients problem in deep neural networks. ResNet-34 uses shortcut connections that aid in carrying the gradients back to the initial layers without the issue of vanishing gradient problem. Thus, it is used as the encoder of the modified U-Net model. Figure 21 and Figure 22 show the proposed architecture of the modified U-Net, and Figure 23 shows the original architecture of U-Net.

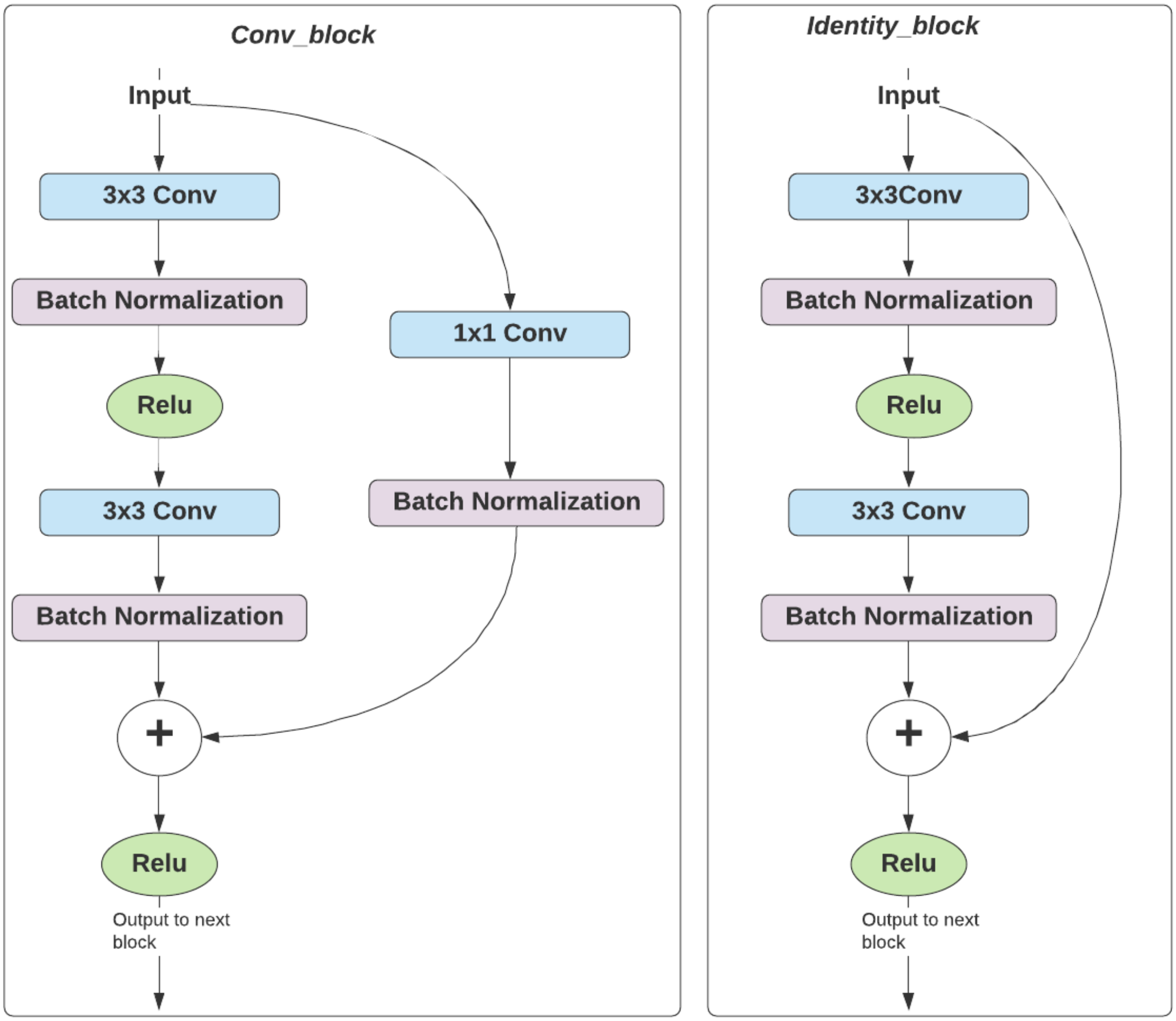


Figure 21: Convolutional and Identity residual blocks for Modified U-Net

The ResNet-34 [95] encoder in the modified U-Net first applies a convolutional layer to the input data with 64 filters (same as the original U-Net) with a filter size of 7×7 pixels. Filter size is the size of the filter that is used to apply the convolutional operation. The output from the above layer is passed through a max-pooling layer with stride two, which downsamples (reduces size) the input. This is followed by multiple convolutional and identity residual blocks, consisting of convolutional, batch normalization, and rectified linear unit (ReLU) layers, as shown in Figure 21 and Figure 22. This is done in accordance with the ResNet-34 architecture as proposed by He *et al.* [95].

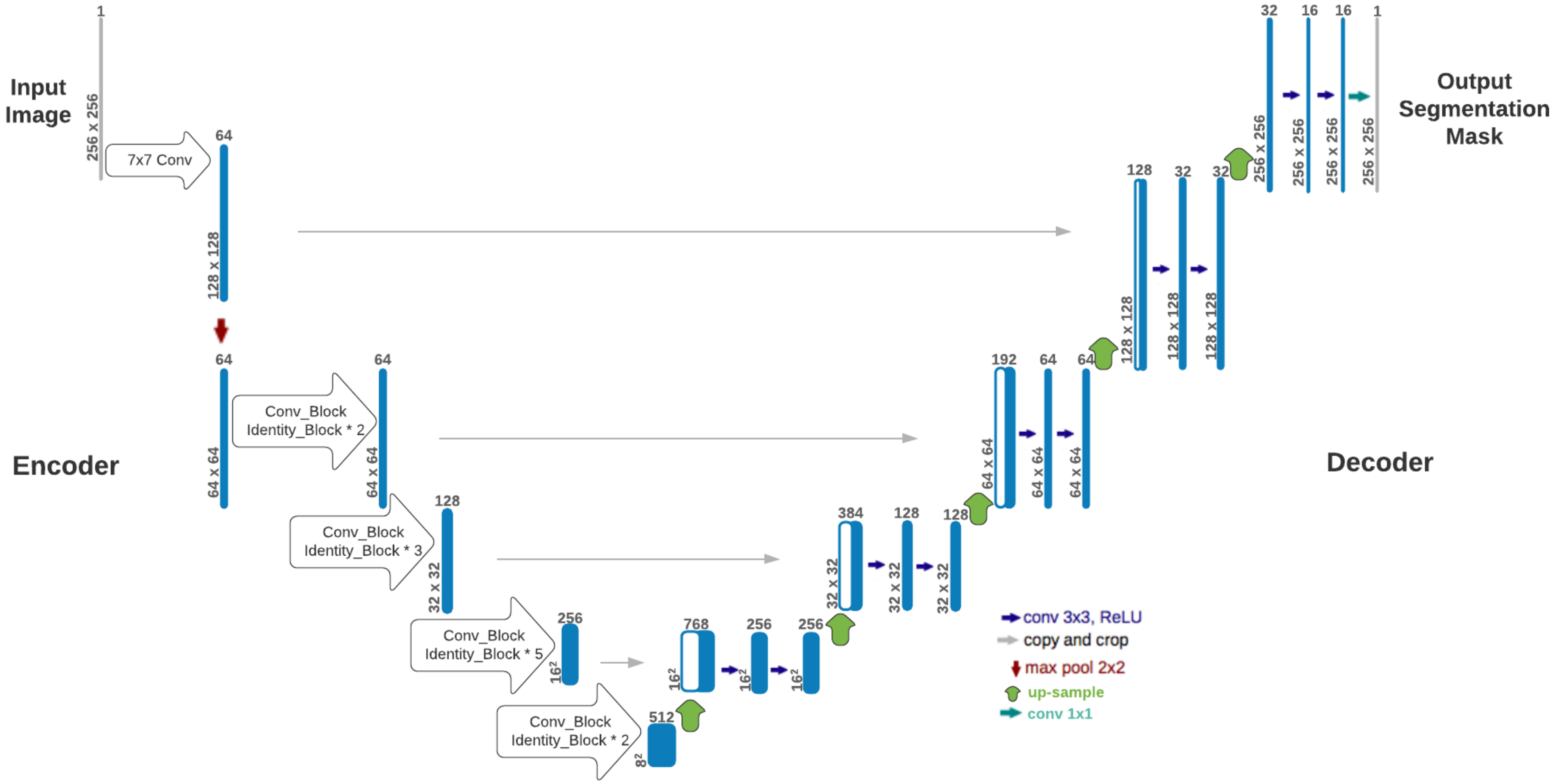


Figure 22: Modified U-Net Architecture

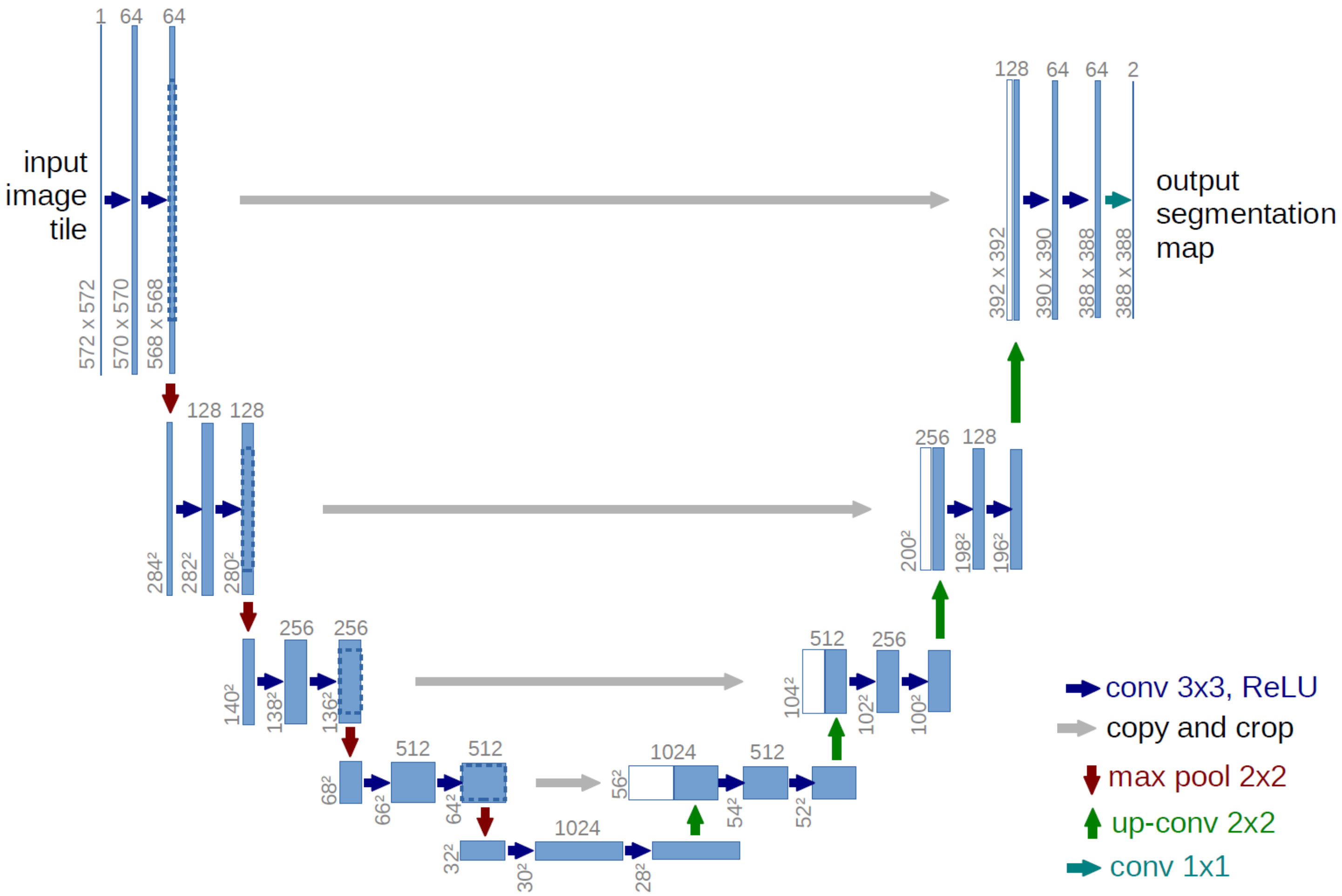


Figure 23: Original U-Net Architecture

The decoder (on the right) follows a similar architecture to the original U-Net. It up-samples the output from the encoder to double the spatial resolution (size) and halves the number of feature channels by using two 3×3 convolutional layers followed by the ReLU activation layer. These are used repetitively to perform up-sampling, and finally, a 1×1 convolutional layer is used to produce an output segmentation mask of the same size as the input image.

To investigate the forgery localization problem, we also make use of other models. One such model is the *LinkNet*. Chaurasia *et al.* [99] proposed LinkNet, a lightweight and fast segmentation network with an intent to utilize the parameters of the neural network more efficiently. The results put forward by Chaurasia *et al.* [99] show that LinkNet is fast and efficient at the task of segmentation and matches and at times exceeds the performance of existing models. Another model used to investigate the forgery localization problem in this thesis is the *PSPNet*. Zhao *et al.* [100] proposed the PSPNet that considered the global context of the image for predicting local level predictions. The PSPNet was the winner of the ImageNet Scene Parsing challenge 2016. Since then, it has been used for segmentation in various other applications such as medical image segmentation [101, 102]. Several experiments were carried out to evaluate the performance of the proposed method and investigate the forgery localization problem. These experiments have been discussed in detail in the next chapter.

Chapter 4

Experimental Results

In this Chapter, extensive experiments are performed to illustrate the efficiency of the IFD-Net and analyze the obstacles in the localization of forged regions. The main goal of these experiments is to study the quality of classification and discuss the reasons behind the difficulties faced in the localization of forged images using the models mentioned in Chapter 3.

4.1 Experimental Setup

In this thesis, all the experiments are carried out on the complete CASIA v2.0 dataset. The dataset consists of 7491 pristine and 5123 forged color images in the JPEG and TIFF formats, ranging from 240 x 160 to 900 x 600 pixels. Out of the 5123 forged images, 1756 forged images are spliced images**,** and the rest are copy-move forged images. All the forged images in the CASIA v2.0 dataset are post-processed to increase detection and localization difficulty. For training the IFD-Net, we use 5/6 of the pristine and forged images, and the remaining 1/6 of these images are used to test the trained model. This data split is done according to the methods in Table 4 so that a valid comparison of their performances can be made. The test set consists of unseen images that are only used to evaluate the performance of the trained model. As there is no validation split of the dataset, we do not tune the hyperparameters and use the standard hyperparameter values for the IFD-Net as mentioned in Section 4.2.

For the localization experiments, we use the IFL-Net along with two other segmentation models LinkNet and PSPNet. As we train these models on the CASIA v2.0 dataset, we divide the dataset into the train, validation, and test sets in the ratio 70:10:20, ensuring that all the models have the same train and test data. Thus, the validation set is used for tuning the hyperparameter values for each of the three models.

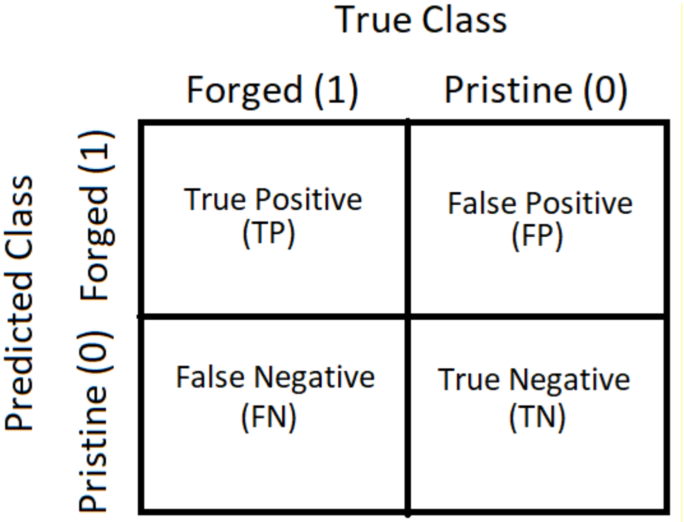
Once both the IFD-Net and the IFL-Net have been trained, they are evaluated on unseen images not part of the training set. Then, they are compared with the existing methods using the accuracy metric for the IFD-Net and the F1 and Intersection over Union (IoU) metrics for the IFL-Net.

All the experiments were performed on a computer with 2.6 GHz Intel(R) Core(TM) i7-10750H CPU with 16 GB RAM and NVIDIA GeForce RTX 2070 GDDR6 @ 8 GB (256 bits) GPU.

4.2 Image Forgery Detection using ResNet-50

Pretrained ResNet-50 model is used in the IFD-Net to detect forged images in the CASIA v2.0 dataset. All the forged and authentic images in the train and test sets are converted from the RGB color channel to the YCbCr color channel. Then only the Cr channel is extracted from these images and passed to the next step for calculating the LBP. For calculating the LBP, the following values as recommended by many other researchers are used for the parameters in Equation 9 in Section 3.3: **p** is the number of neighborhood pixels set to **8**, and **r** is the radius of the neighborhood set to **1** [103, 104]. The generated LBP images are resized to 224 x 224 pixels to maintain the same size over the train and test sets. 224 x 224 image size was chosen to resize all the images as it is the image size originally used by the authors of the ResNet-50 model to train it, and we use this pre-trained ResNet-50 model in the IFD-Net.

The metric used to evaluate the performance is Accuracy. Accuracy is defined in terms of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN), as shown in Equation 11. Figure 24 shows the confusion matrix that defines true positives, false positives, false negatives, and true negatives for forgery detection by IFD-Net.



		True Class	
		Forged (1)	Pristine (0)
Predicted Class	Forged (1)	True Positive (TP)	False Positive (FP)
	Pristine (0)	False Negative (FN)	True Negative (TN)

Figure 24: Confusion Matrix

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

Figure 25 shows the loss and accuracy graphs of the IFD-Net on the CASIA v2.0 dataset. IFD-Net makes use of the pre-trained ResNet-50 model as proposed by He *et al.* [95] along with hyperparameters such as Adam optimizer, learning rate as 0.0000002, and batch size as 32. When training deep neural networks, a batch size of 32 is a good value, as recommended by Bengio [105]. When using a pre-trained model for a given task, it is suggested to use a small learning rate [106]. As the IFD-Net uses a pre-trained ResNet-50 model, we use a small learning rate of 0.0000002. Thus, these standard hyperparameter values are used, and no hyperparameter tuning is performed as there is no validation split in the dataset. Also, binary labels are used for both train and test sets for the IFD-Net as it is a binary classification problem. We make use of ImageDataGenerator class from the Keras library. ImageDataGenerator is an inbuilt method of the Keras library in Python that helps provide binary labels for the images and iterating over the train and test sets. To analyze the impact of geometric augmentations on the IFD-Net, we also experiment by applying various geometric augmentations such as rotation, vertical flipping, and horizontal flipping. Figure 26 shows the loss and accuracy graph of IFD-Net on the CASIA v2.0 dataset when using these geometric augmentations.

Comparing the graphs in Figure 25 and Figure 26, we see that the model trained without geometric augmentations converges faster. When geometric augmentations are used on the data for training IFD-Net, the loss is higher (around 26 %), as seen in Figure 26. Also, using augmentations has no improvement in the detection accuracy of IFD-Net. The accuracy achieved by the model without geometric augmentations is 91.35 %, whereas the accuracy with geometric augmentations is 89.28 %. Using geometric augmentations makes the model take a more significant number of epochs[1] to converge and thus makes it computationally more expensive. Thus, using geometric augmentations such as flipping and rotation for this task on the CASIA v2.0 is not beneficial and is more computationally expensive.

[1] An epoch is one complete pass of the entire dataset through the neural network.

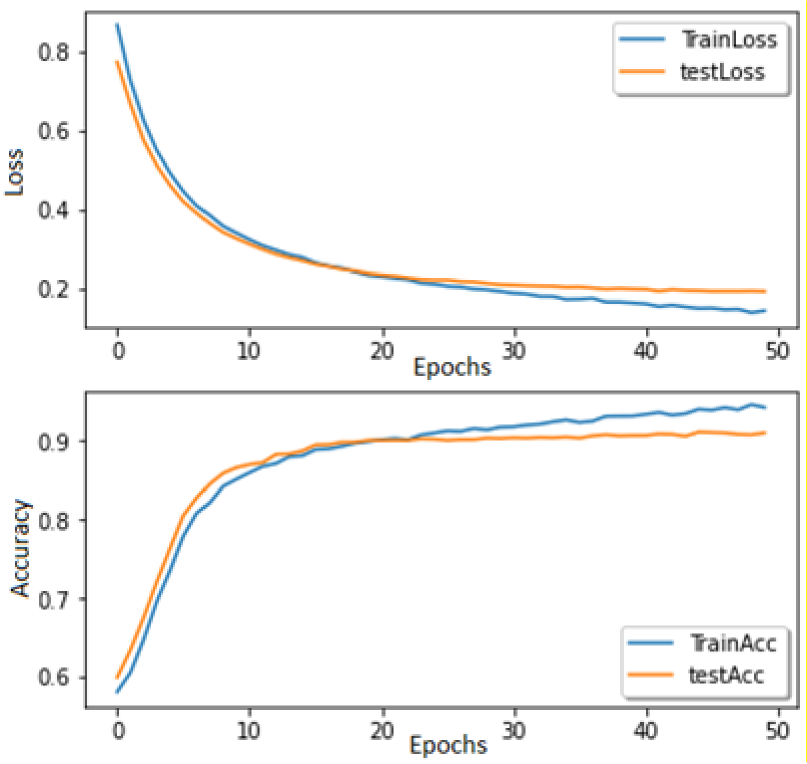


Figure 25: Loss & Accuracy Graph, CASIA v2.0 Dataset

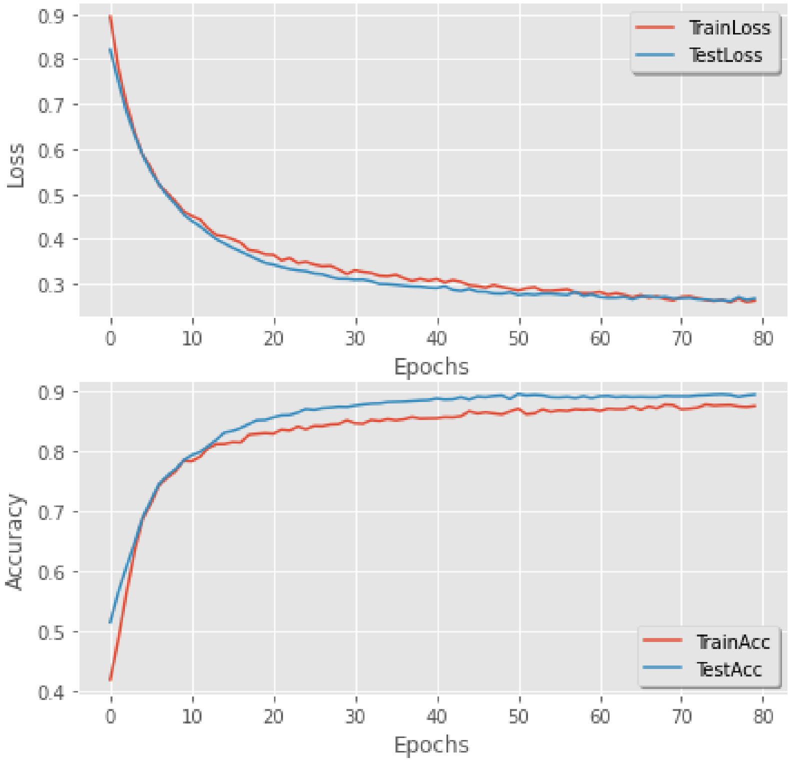


Figure 26: Loss & Accuracy Graphs on CASIA v2.0 dataset with augmentations

Figure 27 shows the impact of RGB, Noiseprint, Cr channel, and LBP images on the detection accuracy of IFD-Net on the CASIA v2.0 dataset. The detection accuracy is the lowest when using RGB images. This is because most image forgers use the RGB color space to forge images and apply post-processing operations to hide the traces of forgery by smoothening the edges of the forged region. Using the noise residual maps generated using Noiseprint does improve the accuracy; however, it is still low. As discussed in Section, Noiseprint captures the camera-specific noise-based fingerprints from images. In spliced images, as the forged region belongs to a different image with a different noise fingerprint, the noise residual map can highlight the difference between the forged region and the original image. However, in copy-move forged images, these camera-specific fingerprints remain the same throughout the image as the copy-move forged region belongs to the same image. Thus, the noise residual map that Noiseprint outputs for a copy-move forged image will be like the noise residual map of its original image as both have the same camera-specific noise-based fingerprint. This can be observed in the Figure 33 in the Appendix. Thus, when these noise residuals maps are used for detecting forged and pristine images, the IFD-Net misclassifies many copy-move forged images as pristine images because the noise residual map does not show any significant variation in the fingerprints in both pristine and copy-move forged images. It can be seen from Figure 27 that the detection accuracy of Cr channel images is much higher than in the previous two cases. The edge sensitivity and sharpness of the forged objects are more noticeable in the chroma channels like the Cr channel. Even though image forgers try to hide and smoothen the edges of forged objects in the RGB color space, some traces of forgery (such as rough edges) are left in the chrominance channels. This helps the IFD-Net to classify images in the CASIA v2.0 dataset as forged or pristine more effectively. The detection accuracy is highest in the case of CR + LBP images. When the LBP operator is applied to each Cr channel image, it sharpens the rough edges of the forged object and the micro-edges inside the forged object in the CR channel images. This makes the edges of the forged object more prominent because of LBP's ability to highlight micro-patterns. Thus, Cr + LBP images perform the best amongst all other types of images used, as shown in Figure 27.

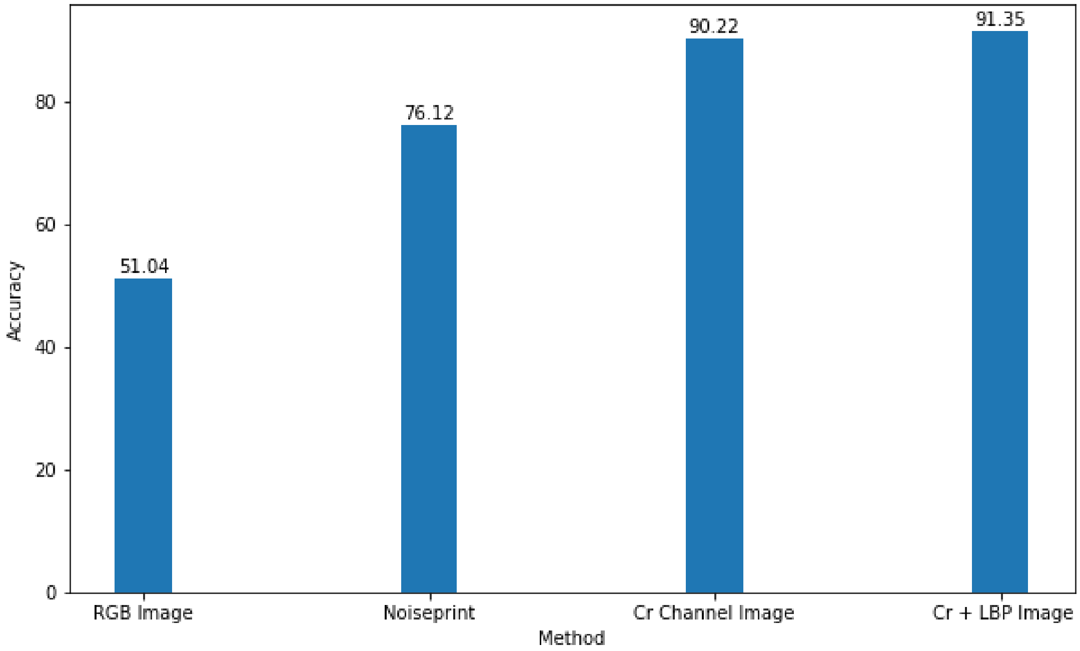


Figure 27: Effect of Noiseprint, Cr Channel and LBP on detection accuracy on CASIA v2.0

We also compare the forgery detection accuracy of the IFD-Net with other methods on the CASIA v2.0 dataset. Table *4* provides this comparison of IFD-Net with other methods. None of these methods that we compare use a pre-trained model. The IFD-Net performs better than the methods proposed by Sutthiwan *et al.* [107] and Zhongwei *et al.* [108]. However, its detection accuracy is just a little less than the method proposed by Li *et al.* [109].

Table 4: Detection Accuracy Comparison with other methods on CASIA v2.0 dataset

Method	*Feature Extractor*	*Average Test Accuracy (%)*
Sutthiwan et al. [107]	Markovian Rate Transform	79.74
Zhongwei et al. [108]	Markov + DCT + DWT	89.76
Li et al. [109]	Markov + QDCT	92.38
IFD-Net v1	**Cr + LBP + ResNet-50**	**91.35**

The performance of IFD-Net v1 is at par with all the other techniques it is compared with. Also, the IFD-Net v1, which uses Cr + LBP images, performs better than IFD-Net v2, which uses the noise residual maps extracted using the Noiseprint model. Several experiments were conducted to analyze the localization of forged regions in the forged images present in the CASIA v2.0 dataset that are discussed in the next section.

4.3 Analysis of Forgery Localization

The IFL-Net and other deep learning models such as LinkNet and PSPNet have been used to investigate the localization of forged regions in forged images, as discussed in Section 3.7. Like the forgery detection method where the Cr channel is extracted and LBP is applied, the forgery localization method follows the same procedure. The same parameters are used for calculating the LBP as used in IFD-Net.

The IFL-Net is used along with the hyperparameters, such as the initial batch size of 32, the Adam optimizer, and the learning rate set to 0.0007. These hyperparameters were chosen after tuning the hyperparameters on the validations set, as shown in Table 7 in the Appendix. The images are resized to 256 x 256 pixels to maintain the same size over the train and test sets for IFL-Net and LinkNet. However, as PSPNet accepts image sizes which are a multiple of 48, the images are resized to 240 x 240 pixels for the PSPNet experiments.

The performance metrics used to evaluate the performance of the models mentioned above are F_1 score and Intersection over Union (IoU). The harmonic mean of recall and precision is known as F_1 score. First, the True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) are calculated to calculate the recall and precision, which are then further used to calculate the F1 score. In addition, the metric IoU, also known as the Jaccard Index, is used to quantify the percent of overlap between the ground-truth mask and the predicted mask. From Equation 16 and 19, we can see that both the F1 score and IoU look similar. In comparison to the F1 score, IoU penalizes individual instances of misclassifications more than the F1 score. The F1 score tends to provide a measure closer to the average performance as it is the weighted average of precision and recall. IoU and the F1 score are always within a factor of 2 of each other, as shown in Equation 20. The terms y_i^{true} and y_i^{pred} in the following equations refer to the ground truth labels and the predicted labels, respectively.

$$TP = \sum_{i=1}^{n} y_i^{true} * y_i^{pred} \tag{12}$$

$$TN = \sum_{i=1}^{n} (1 - y_i^{true}) * \left(1 - y_i^{pred}\right) \tag{13}$$

$$FP = \sum_{i=1}^{n} (1 - y_i^{true}) * \ y_i^{pred} \tag{14}$$

$$FN = \sum_{i=1}^{n} y_i^{true} * \left(1 - y_i^{pred}\right) \tag{15}$$

$$F_1 \ = 2 \cdot \frac{precision \cdot recall}{precision + recall} = \frac{2TP}{2TP + FP + FN} \ = \ Dice\ Coefficient \tag{16}$$

Where,

$$precision \ = \frac{TP}{TP + FP} \tag{17}$$

$$recall \ = \frac{TP}{TP + FN} \tag{18}$$

$$IoU = \frac{intersection}{union} = \frac{TP}{TP + FP + FN} \tag{19}$$

$$\frac{F1}{2} \leq \ IoU \ \leq F1 \tag{20}$$

The forged images in the CASIA v2.0 dataset are highly class imbalanced, as discussed in Section 2.5. As discussed in Section 2.6, various loss functions such as dice loss and weighted binary-cross entropy loss are used to give more importance to the minority class. Figure 28 and Figure 29 show the loss and F1 score graphs of IFL-Net when using the weighted binary-cross entropy loss. These graphs show that the IFL-Net overfits the data while training and thus will not generalize well.

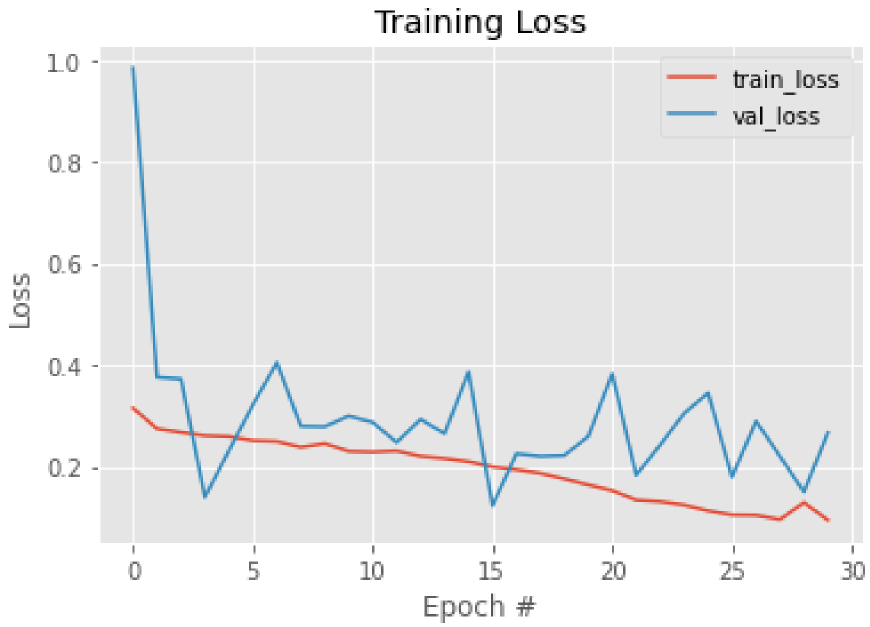


Figure 28: IFL-Net Training & Validation Loss Graph, CASIA v2.0 Dataset

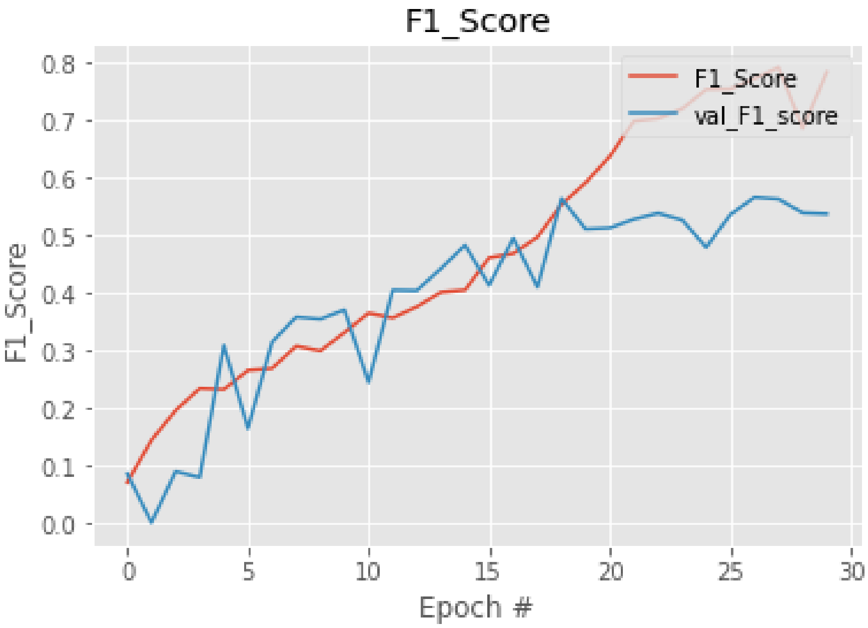


Figure 29:IFL- Net Training & Validation F1 score Graph, CASIA v2.0 Dataset

However, to further investigate this issue, we also conduct experiments using other models like LinkNet and PSPSNet. Figure 30 and Figure 31 show the F1 score graphs of the models LinkNet and PSPNet when using the WBCE Loss. It can be observed from these graphs that even these models are overfitting. We also compare the localization performances of the IFL-Net, LinkNet, and PSPNet while using the WBCE and Dice loss functions. Table 5 and Table 6 compare their performance on the test data. These tables show that all three models do not localize well, as represented by the low F1 and IoU scores.

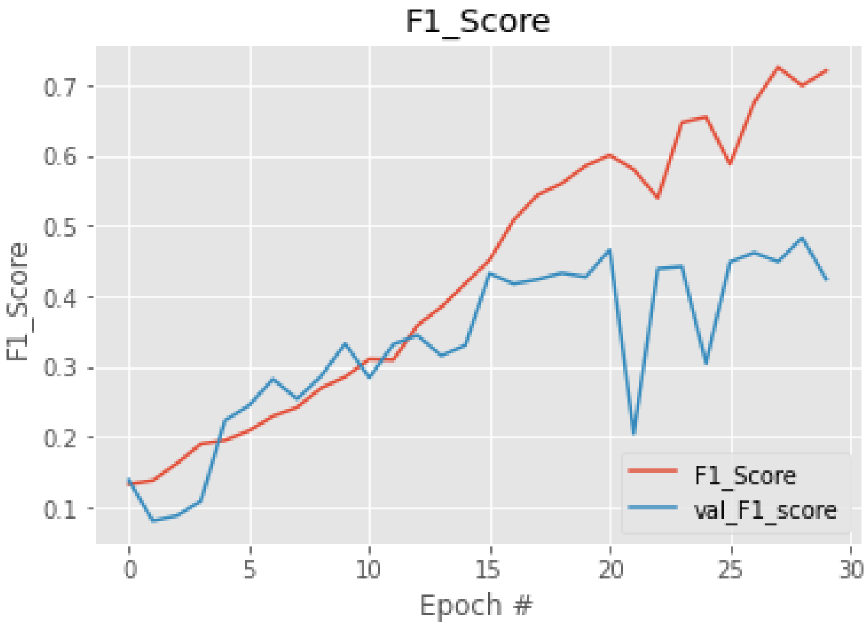


Figure 30: LinkNet Training & Validation F1 score graph

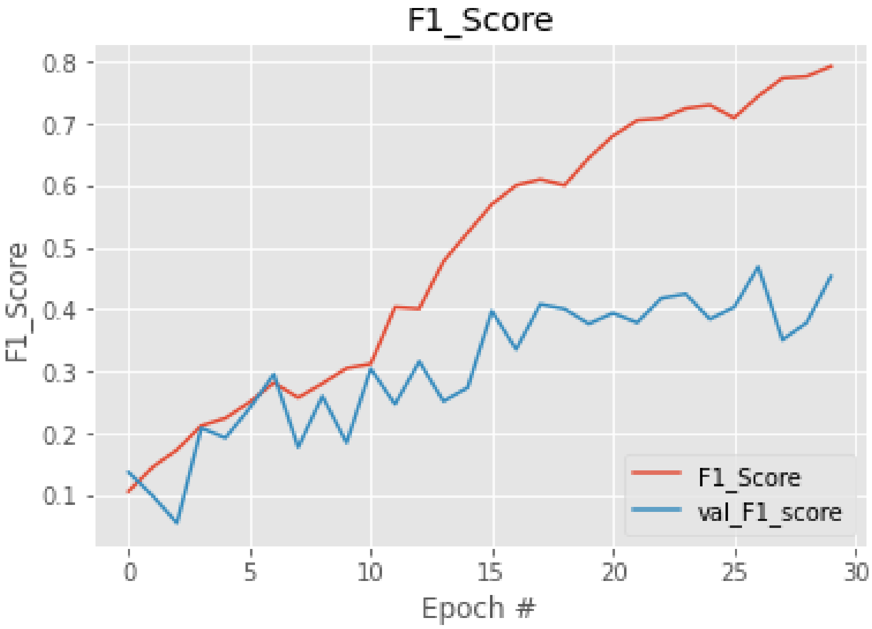


Figure 31: PSPNet Training & Validation F1 score graph

Table 5: Localization Model Comparison, WBCE Loss

Model	*Test F_1 Score*	*Test IoU*
LinkNet	0.4316	0.2828
PSPNet	0.3419	0.2148
IFL-Net	0.4328	0.2855

Table 6: Localization Model Comparison, Dice Loss

Model	*Test F_1 Score*	*Test IoU*
LinkNet	0.4562	0.3025
PSPNet	0.4399	0.2909
IFL-Net	0.4340	0.2858

Several attempts to adjust the hyperparameters, such as the learning rate, were carried to adjust the training of these models. However, no significant improvement was observed, as shown in Table 8 and Table 9. One reason for the overfitting of all these models is the high-class imbalance in the images of the CASIA v2.0 dataset. When there is a class imbalance in training data, learners are more likely to overclassify the majority class because of its higher prior probability. The high imbalance caused by the majority class leads to the majority class accounting for most of the model's loss. As a result, pixels of the minority class are misclassified more frequently than pixels of the majority class. Using the dice loss and weighted binary cross-entropy loss functions that focus more on the minority class show only a slight improvement, as seen from the results in Table 5 and Table 6. A recent study by Li *et al.* [110] shows that it is not difficult for a neural network to predict the minority classes without benefitting from these loss functions during training due to overfitting. It is observed that the above-trained models did overfit and, in fact, in many cases overclassified the majority class, thus misclassifying the minority class. Figure 34 in the Appendix shows that the trained model overclassifies the majority class and misclassifies the minority class. Thus, we can conclude that deep learning-based image segmentation models such as the modified U-Net, LinkNet, and PSPNet do not perform well at image forgery localization on the CASIA v2.0 dataset because of the high-class imbalanced data. Also, further work on handling highly imbalanced data by deep learning models is required, which can help solve such localization problems in various domains.

The following section intends to provide an insight into the overall performance of both the IFD-Net and the IFL-Net, along with the possible improvements that can be made.

4.4 Discussion

One can observe from the above-mentioned experimental results in Table 4 that the IFD-Net performs better on the CASIA v2.0 dataset compared to most of the related works it is compared with. Two versions of IFD-Net were used for the experiments. The first version used LBP to extract features, whereas the second version used Noiseprint to extract the noise residual maps that were then fed to the ResNet-50 model. IFD-Net v1 performs better than IFD-Net v2 for detecting forged images.

Although IFD-Net has performed well, further experimentation is required to get more information about its performance on other image forgery datasets. Analyzing the forgery localization problem in the CASIA v2.0 dataset, it is observed that high-class imbalance in the forged images causes poor performance of various deep learning-based models. Even using loss functions that focus more on the minority class only provides a slight improvement in the performance of the models, as shown in Table 5 and Table 6 above. As mentioned by many other researchers, deep learning with class imbalanced data is an understudied topic, and more work in this direction is required.

To summarize it all, we can say that the use of Cr + LBP images for forgery detection helped the IFD-Net perform well on unseen images compared to other image preprocessing methods, as shown in Figure 27. Also, further studies in the direction of class imbalanced data are required to find better solutions to solve the issue of class imbalance with deep learning models. It is evident from the experimental results that the IFD-Net and analysis of the forgery localization in the CASIA v2.0 dataset address all the research objectives considered at the start of this thesis.

Chapter 5

Conclusion and Future Work

In this chapter, we conclude this thesis by justifying all the research objectives listed in Chapter 1 and providing an insight into future work.

5.1 Conclusion

There are many challenges to overcome when implementing a solution for image forgery detection and localization. Existing solutions based on deep learning do provide decent forgery detection, but their performance is not satisfactory for forgery localization. With the recent advancements in powerful computers and the rise of deep learning, it has been possible to use various deep neural network architectures in varied domains. However, deep learning in the context of imbalanced class datasets is still understudied. Thus, this thesis presents a texture-based solution for image forgery detection and analyzes the performance of various deep learning models at the task of localization of forged images.

The main objective of this thesis was to provide an image forgery detection method that can efficiently differentiate forged images from pristine images. Moreover, in accordance with Section 1.3, the following objectives were successfully obtained:

- Proposing a well-structured image forgery detection architecture based on textural features using a pre-trained ResNet-50 model. The steps of our methodology to achieve this objective were presented in Section 3.6. With the help of these steps, we explain how we use the textural features using LBP in classifying the images in the CASIA v2.0 dataset as forged or pristine using a pre-trained ResNet-50 model. We also explore the avenue of utilizing the noise residual maps of the images using Noiseprint as an input to the pre-trained ResNet-50 model for image forgery detection.
- Using an ablation study in Section 4.2, we show the positive impact of using LBP at image forgery detection when compared with other image preprocessing methods. This is used to show how can the textural inconstancies introduced in an image because of the forgery operation be used to classify images as forged or pristine.

- Analyzing the performance of the modified U-Net architecture and other deep learning models like LinkNet and PSPNet to investigate why these deep learning models face difficulties in localizing the forged regions in the forged images of the CASIA v2.0 dataset. Section 4.3 puts forward the reasons behind the performance of these models at the task of forgery localization and suggests that more study in this direction is required.

In summary, this thesis presents an image forgery detection and localization method based on textural features and deep learning models. The IFD-Net can be used in various applications to distinguish between a forged image and a pristine image. This method can help reduce the spread of fake news and fake images that intend to change public opinion. Several experiments are carried out on the benchmarked dataset CASIA v2.0 to investigate why various deep learning models find it difficult to localize the forged regions in the forged images.

5.1 Future Work

In addition to the work presented in this thesis, the following directions can be explored as future work:

- To study how the performance of IFD-Net can be further improved at detecting images as forged of pristine. This can help improve the ability of IFD-Net to detect different kinds of image forgeries and further prevent the spread of fake news and misinformation.
- Deep learning from class imbalanced data is still understudied, and further work in this direction can be beneficial for applying deep learning in various domains. Thus, further studying class imbalance in datasets to devise solutions that can improve the performance of deep learning models on highly imbalanced datasets.

Appendix

Figure 32 shows the loss and accuracy plots of AlexNet for classifying images of the CASIA v2.0 dataset as forged or pristine.

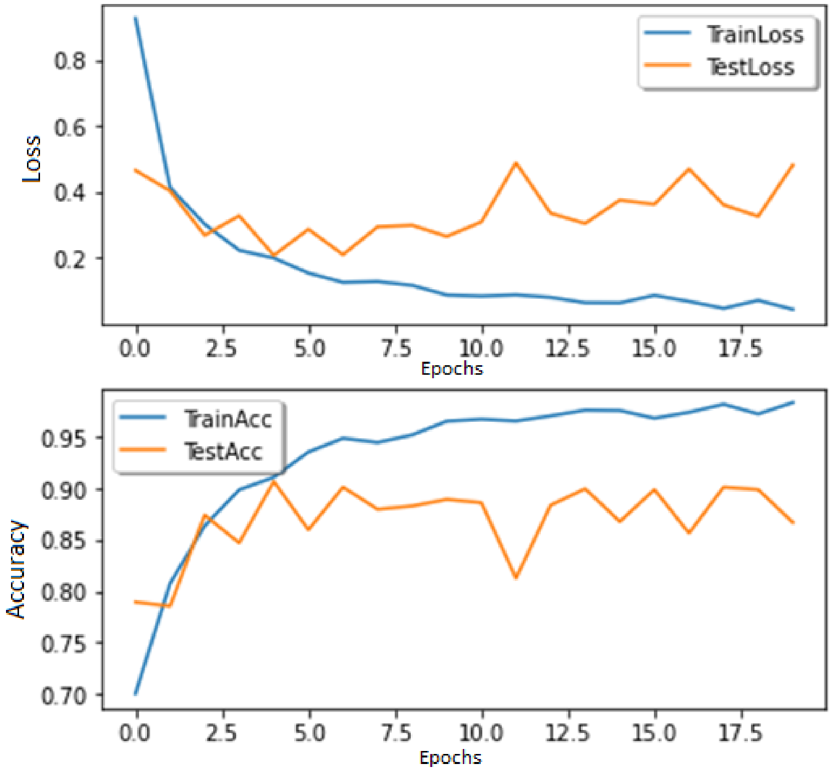
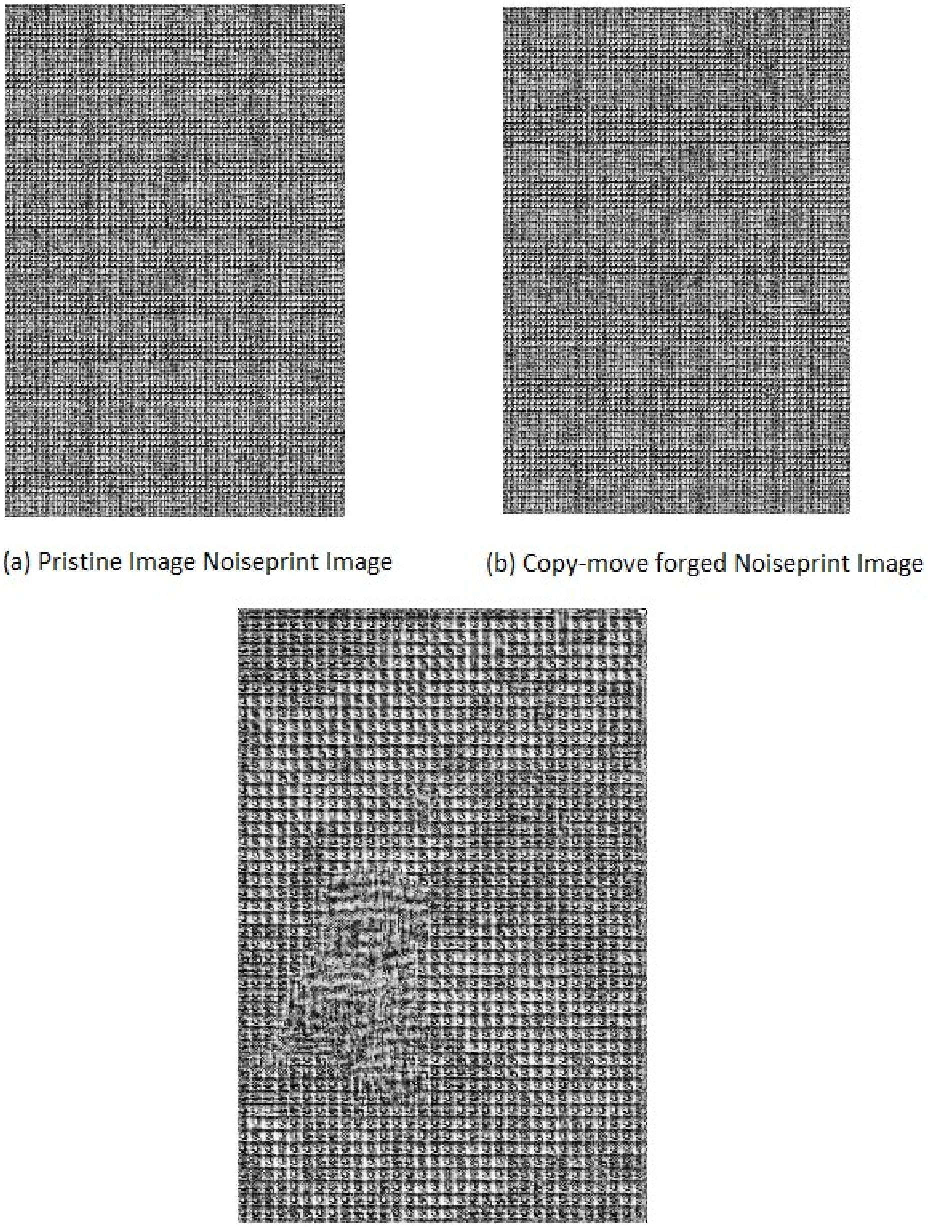


Figure 32: AlexNet performance on CASIA v2.0



(a) Pristine Image Noiseprint Image

(b) Copy-move forged Noiseprint Image

(c) Spliced Noiseprint Image

Figure 33: Noiseprint Image of pristine and forged images

Table 7: Some hyperparameter settings experimented for IFL-Net

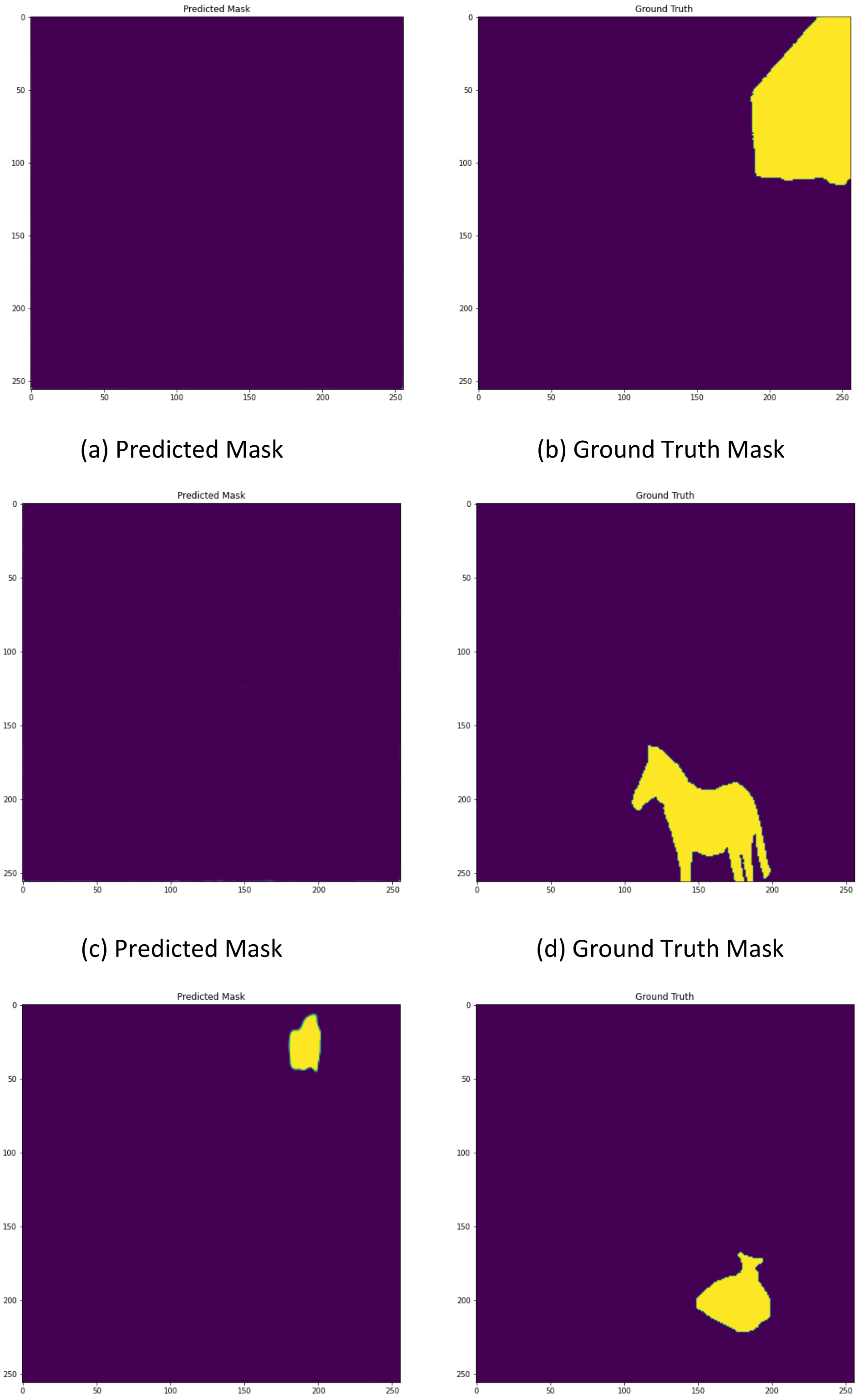
Input Size	*Loss Function*	*Learning Rate*	*Validation F_1 Score*	*Validation IoU*
256 x 256	WBCE	0.0007	0.5496	0.3835
256 x 256	WBCE	0.00001	0.2437	0.1204
256 x 256	Dice Loss	0.0007	0.5027	0.3411

Table 8: Some hyperparameter settings experimented for LinkNet

Input Size	*Loss Function*	*Learning Rate*	*Validation F_1 Score*	*Validation IoU*
256 x 256	WBCE	0.0007	0.4829	0.3231
256 x 256	WBCE	0.00001	0.2896	0.1437
256 x 256	Dice Loss	0.00007	0.3504	0.2167
256 x 256	Dice Loss	0.0007	0.5041	0.3454

Table 9:Some hyperparameter settings experimented for PSPNet

Input Size	*Loss Function*	*Learning Rate*	*Validation F_1 Score*	*Validation IoU*
240 x 240	WBCE	0.00001	0.1673	0.1232
240 x 240	WBCE	0.0007	0.4080	0.2617
240 x 240	Dice Loss	0.0007	0.5483	0.3853
240 x 240	Dice Loss	0.00001	0.3371	0.1958



(a) Predicted Mask

(b) Ground Truth Mask

(c) Predicted Mask

(d) Ground Truth Mask

(e) Predicted Mask

(f) Ground Truth Mask

Figure 34: IFL-Net Forgery Localization Predicitons

References

[1] R. Eveleth, "How fake images change our memory and behaviour," BBC, 12 12 2012. [Online]. Available: https://www.bbc.com/future/article/20121213-fake-pictures-make-real-memories. [Accessed 14 04 2021].

[2] H. Farid, "Image forgery detection," *IEEE Signal Processing Magazine,* vol. 26, no. 2, pp. 16 - 25, 2009.

[3] C. d. Kaur and N. Kanwal, "An Analysis of Image Forgery Detection Techniques," *Statistics, Optimization & Information Computing,* vol. 7, no. 2, pp. 486-500, 2019.

[4] M. Kharang and A. Doegar, "Copy-Move Forgery Detection Methods: A Critique," *Advances in Information Communication Technology and Computing,* vol. 135, pp. 501-523, 2021.

[5] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Hippolyte_Bayard. [Accessed 27 01 2021].

[6] A. Kashyap, R. S. Parmar, M. Agarwal and H. Gupta, "An Evaluation of Digital Image Forgery Detection Approaches," *International Journal of Applied Engineering Research,* vol. 12, no. 15, p. 4747–4758, 2017.

[7] P. Bhole and D. Wajgi, "An Approach for Image Forgery Detection," *International Journal of Research in Engineering, Science and Management,* vol. 3, no. 2, p. , 2020.

[8] V. Christlein, C. Riess, J. Jordan, C. Riess and E. Angelopoulou, "An Evaluation of Popular Copy-Move Forgery Detection Approaches," *IEEE Transactions on Information Forensics and Security,* vol. 7, no. 6, pp. 1841 - 1854, 2012.

[9] J. Fridrich, D. Soukal and J. Lukáš, "Detection of Copy-Move Forgery in Digital Images," in *Proceedings of Digital Forensic Research Workshop*, 2003.

[10] M. NIZZA and P. J. LYONS, "The New York Times," 10 07 2008. [Online]. Available: https://thelede.blogs.nytimes.com/2008/07/10/in-an-iranian-image-a-missile-too-many/. [Accessed 28 01 2021].

[11] R. Thebault, "The Washington Post," [Online]. Available: https://www.washingtonpost.com/politics/2020/01/06/gop-congressman-tweeted-fake-image-obama-with-iranian-president-they-never-met/. [Accessed 28 01 2021].

[12] X. Pan, X. Zhang and S. Lyu, "Exposing Image Splicing with Inconsistent Local Noise Variances," in *IEEE International Conference on Computational Photography (ICCP)*, Seattle, 2012.

[13] S. Singhal and V. Ranga, "Passive Authentication Image Forgery Detection Using Multilayer CNN," *Mobile Radio Communications and 5G Networks,* vol. 140, pp. 237-249, 2021.

[14] J. Puckett, "WUSA9," [Online]. Available: https://www.wusa9.com/article/news/verify/verify-fake-photo-of-obama-and-iranian-president-posted-by-congressman/507-3ddebd2a-b83a-4d4c-9b3c-64d3f2f2927d. [Accessed 28 01 2021].

[15] "David Oswald Photography," [Online]. Available: http://www.davidoswaldphotography.com/photo-retouching.html#:~:text=Technical%20retouching%20is%20photo%20restoration,and%20interesting%20images%20for%20advertisements.. [Accessed 28 01 2021].

[16] J. Lategan, "Digital Image Retouching," Joe Lategan, [Online]. Available: http://www.joelategan.com/post-processing.php. [Accessed 28 01 2021].

[17] E. Hall, "People Are Sharing Photoshopped Pictures Of Trump — Again," BuzzFeed News, [Online]. Available: https://www.buzzfeednews.com/article/ellievhall/trump-viral-photoshop-fat-fake-twitter. [Accessed 29 01 2021].

[18] N. M. AlShariah and A. K. J. Saudagar, "Detecting Fake Images on Social Media using Machine Learning," *International Journal of Advanced Computer Science and Applications,* vol. 10, no. 12, pp. 170-176, 2019.

[19] S. K. Mankar and P. D. A. A. Gurjar, "Image Forgery Types and Their Detection: A Review," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 5, no. 4, pp. 174-178, 2015.

[20] P. Zhou, X. Han, V. I. Morariu and L. S. Davis, "Learning Rich Features for Image Manipulation Detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018.

[21] S. J. ,. D. R. K. B. Varsha Sharma, "Image Forgery and it’s Detection Technique: A Review," *International Research Journal of Engineering and Technology (IRJET),* vol. 03, no. 03, pp. 756-762, 2016.

[22] X. Lin, J.-H. Li, S.-L. Wang, A.-W.-C. Liew, F. Cheng and X.-S. Huang, "Recent Advances in Passive Digital Image Security Forensics: A Brief Review," *Engineering,* vol. 4, no. 1, pp. 29-39, 2018.

[23] M. A. Qureshi and M. Deriche, "A bibliography of pixel-based blind image forgery detection techniques," *Signal Processing: Image Communication,* vol. 39, no. Part A, pp. 46-74, 2015.

[24] R. Agarwal, D. Khudaniya, A. Gupta and K. Grover, "Image Forgery Detection and Deep Learning Techniques: A Review," in *International Conference on Intelligent Computing and Control Systems*, Madurai, India, 2020.

[25] A. Piva, M. Barni, F. Bartolini and V. Cappellini, "DCT-based Watermark Recovering without Resorting to the Uncorrupted Original Image," in *International Conference on Image Processing*, Santa Barbara, 1997.

[26] R. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM,* vol. 21, no. 2, p. 120–126, 1978.

[27] O. M.Al-Qershi and B. EeKhoo, "Passive detection of copy-move forgery in digital images: State-of-the-art," *Forensic Science International,* vol. 281, no. 1 - 3, pp. 284 - 295, 2013.

[28] X. Yu, C. Wang and X. Zhou, "Review on Semi-Fragile Watermarking Algorithms for Content Authentication of Digital Images," *Future Internet,* vol. 9, no. 4, 2017.

[29] A. Tefas, N. Nikolaidis and I. Pitas, "Image Watermarking: Techniques and Applications," in *The Essential Guide to Image Processing*, Academic Press, 2009, pp. 597-648.

[30] E. A. A. Vega, E. G. Fernández, A. L. S. Orozco and L. J. G. Villalba, "Passive Image Forgery Detection Based on the Demosaicing Algorithm and JPEG Compression," *IEEE Access ,* vol. 8, pp. 11815 - 11823, 2020.

[31] A. C. Popescu and H. Farid, "Exposing Digital Forgeries by Detecting Duplicated Image Regions," Dartmouth Digital Commons, 2004.

[32] D. Cozzolino, G. Poggi and L. Verdoliva, "Efficient Dense-Field Copy–Move Forgery Detection," *IEEE Transactions on Information Forensics and Security,* vol. 10, no. 11, pp. 2284 - 2297, 2015.

[33] X. Kang and S. Wei, "Identifying Tampered Regions Using Singular Value Decomposition in Digital Image Forensics," in *International Conference on Computer Science and Software Engineering*, Hubei, China, 2008.

[34] H. Huang, W. Guo and Y. Zhang, "Detection of Copy-Move Forgery in Digital Images Using SIFT Algorithm," in *IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, Wuhan, China, 2008.

[35] A. R. H. Khayeat, X. Sun and P. L. Rosin, "Improved DSIFT Descriptor Based Copy-Rotate-Move Forgery Detection," *Lecture Notes in Computer Science,* vol. 9431, pp. 642-655, 2015.

[36] X. Bo, W. Junwen, L. Guangjie and D. Yuewei, "Image Copy-Move Forgery Detection Based on SURF," in *International Conference on Multimedia Information Networking and Security*, Nanjing, Jiangsu, China, 2010.

[37] I. Amerini, L. Ballan, R. Caldelli, A. D. Bimbo and G. Serra, "A SIFT-Based Forensic Method for Copy–Move Attack Detection and Transformation Recovery," *IEEE Transactions on Information Forensics and Security ,* vol. 6, no. 3, pp. 1099 - 1110, 2011.

[38] G. Muhammad, M. Hussain and G. Bebis, "Passive copy move image forgery detection using undecimated dyadic wavelet transform," *Digital Investigation,* vol. 9, no. 1, pp. 49-57, 2012.

[39] J. Zhao and J. Guo, "Passive forensics for copy-move image forgery using a method based on DCT and SVD," *Forensic Science International,* vol. 233, no. 1-3, pp. 158-166, 2013.

[40] C.-S. Park and J. Y. Choeh, "Fast and robust copy-move forgery detection based on scale-space representation," *Multimedia Tools and Applications,* vol. 77, p. 16795–16811, 2018.

[41] J. Zhang, W. Zhu, B. Li, W. Hu and J. Yang, "Image Copy Detection Based on Convolutional Neural Networks," *Communications in Computer and Information Science,* vol. 663, pp. 111-121, 2016.

[42] Y. Zhang, J. Goh and V. L. L. T. Lei Lei Win, "Image Region Forgery Detection: A Deep Learning Approach," *Cryptology and Information Security Series,* vol. 14, pp. 1 - 11, 2016.

[43] Y. Wu, W. Abd-Almageed and P. Natarajan, "BusterNet: Detecting Copy-Move Image Forgery with Source/Target Localization," *Lecture Notes in Computer Science,* vol. 11210, pp. 170-186, 2018.

[44] B. Bayar and M. C. Stamm, "Constrained Convolutional Neural Networks: A New Approach Towards General Purpose Image Manipulation Detection," *IEEE Transactions on Information Forensics and Security,* vol. 13, no. 11, pp. 2691 - 2706, 2018.

[45] J. Zhou, J. Ni and Y. Rao, "Block-Based Convolutional Neural Network for Image Forgery Detection," *Kraetzer C., Shi YQ., Dittmann J., Kim H. (eds) Digital Forensics and Watermarking. IWDW 2017. Lecture Notes in Computer Science,* vol. 10431, pp. 65-76, 2017.

[46] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, Abu Dhabi, United Arab Emirates, 2016.

[47] Y. Liu, Q. Guan and X. Zhao, "Copy-move forgery detection based on convolutional kernel network," *Multimedia Tools and Applications ,* vol. 77, p. 18269–18293, 2018.

[48] H. Zhou, Y. Shen, X. Zhu, B. Liu, Z. Fu and N. Fan, "Digital image modification detection using color information and its histograms," *Forensic Science International,* vol. 266, pp. 379-388, 2016.

[49] T. Pomari, G. Ruppert, E. Rezende, A. Rocha and T. Carvalho, "Image Splicing Detection Through Illumination Inconsistencies and Deep Learning," in *IEEE International Conference on Image Processing (ICIP)*, Athens, Greece, 2018.

[50] W. Chen, C. Chen and Y. Shi, "A natural image model approach to splicing detection," in *Proceedings of the 9th Workshop on Multimedia & security*, Dallas, Texas, 2007.

[51] J. Wang, Q. Ni, G. Liue, X. Luod and S. Kr.Jha, "Image splicing detection based on convolutional neural network with weight combination strategy," *Journal of Information Security and Applications,* vol. 54, p. 102523, 2020.

[52] S. Lyu, X. Pan and X. Zhang, "Exposing Region Splicing Forgeries with Blind Local Noise Estimation," *International Journal of Computer Vision,* vol. 110, p. 202–221, 2014.

[53] D. Cozzolino, G. Poggi and L. Verdoliva, "Splicebuster: A new blind image splicing detector," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, Rome, Italy, 2016.

[54] M. Chen, J. Fridrich, M. Goljan and J. Lukas, "Determining Image Origin and Integrity Using Sensor Noise," *IEEE Transactions on Information Forensics and Security,* vol. 3, no. 1, pp. 74 - 90, 2008.

[55] G. Chierchia, G. Poggi, C. Sansone and L. Verdoliva, "A Bayesian-MRF Approach for PRNU-Based Image Forgery Detection," *IEEE Transactions on Information Forensics and Security ,* vol. 9, no. 4, pp. 554 - 567, 2014.

[56] C.-T. Li and Y. Li, "Color-Decoupled Photo Response Non-Uniformity for Digital Image Forensics," *IEEE Transactions on Circuits and Systems for Video Technology ,* vol. 22, no. 2, pp. 260 - 271, 2012.

[57] H. Farid and S. Lyu, "Higher-order Wavelet Statistics and their Application to Digital Forensics," in *Conference on Computer Vision and Pattern Recognition Workshop*, Madison, Wisconsin, USA, 2003.

[58] D. Fu, Y. Q. Shi and W. Su, "Detection of Image Splicing Based on Hilbert-Huang Transform and Moments of Characteristic Functions with Wavelet Decomposition," *International Workshop on Digital Watermarking, Lecture Notes in Computer Science,* vol. 4283, pp. 177-187, 2006.

[59] W. Chen, Y. Q. Shi and W. Su, "Image splicing detection using 2-D phase congruency and statistical moments of characteristic function," *Security, Steganography, and Watermarking of Multimedia Contents IX,* vol. 6505, pp. 65050R-1 - 65050R-8, 2007.

[60] C. Gu-chuna, S. Bob, W. Shi-linb and L. Sheng-honga, "Blind Detection of Splicing Image Based on Gray Level Co-occurrence Matrix of Image DCT Domain," *Journal of Shanghai Jiaotong University,* vol. 45, no. 10, pp. 1547-1551, 2011.

[61] E. Kee, M. K. Johnson and H. Farid, "Digital Image Authentication From JPEG Headers," *IEEE Transactions on Information Forensics and Security,* vol. 6, no. 3, pp. 1066 - 1075, 2011.

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

[63] T. Bianchi and A. Piva, "Detection of Nonaligned Double JPEG Compression Based on Integer Periodicity Maps," *IEEE Transactions on Information Forensics and Security,* vol. 7, no. 2, pp. 842 - 848, 2012.

[64] T. Bianchi and A. Piva, "Image Forgery Localization via Block-Grained Analysis of JPEG Artifacts," *IEEE Transactions on Information Forensics and Security,* vol. 7, no. 3, pp. 1003 - 1017, 2012.

[65] H. Farid, "Exposing Digital Forgeries From JPEG Ghosts," *IEEE Transactions on Information Forensics and Security,* vol. 4, no. 1, pp. 154 - 160, 2009.

[66] E. Ramadhani, "Photo splicing detection using error level analysis and laplacian-edge detection plugin on GIMP," *Journal of Physics: Conference Series,* vol. 1193, 2019.

[67] P. Niyishaka and C. Bhagvati, "Image splicing detection technique based on Illumination-Reflectance model and LBP," *Multimedia Tools and Applications,* vol. 80, p. 2161–2175, 2021.

[68] A. K. Jaiswal and R. Srivastava, "Image Splicing Detection using Deep Residual Network," in *International Conference on Advanced Computing and Software Engineering (ICACSE)*, Sultanpur, UP, India, 2019.

[69] Y. Rao, J. Ni and H. Xie, "Multi-semantic CRF-based attention model for image forgery detection and localization," *Signal Processing,* vol. 183, p. 108051, 2021.

[70] R. Salloum, Y. Ren and C.-C. Kuo, "Image Splicing Localization using a Multi-task Fully Convolutional Network (MFCN)," *Journal of Visual Communication and Image Representation,* vol. 51, pp. 201-209, 2018.

[71] N. T. Pham, J.-W. Lee, G.-R. Kwon and C.-S. Park, "Hybrid Image-Retrieval Method for Image-Splicing Validation," *Symmetry,* vol. 11, no. 1, 2018.

[72] "Columbia Image Splicing Detection Evaluation Dataset," [Online]. Available: https://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/AuthSplicedDataSet.htm. [Accessed 17 01 2021].

[73] "Columbia Uncompressed Image Splicing Detection Evaluation Dataset," [Online]. Available: https://www.ee.columbia.edu/ln/dvmm/downloads/authsplcuncmp/. [Accessed 17 01 2021].

[74] "IEEE IFS-TC Image Forensics Challenge Image Corpus," [Online]. Available: http://web.archive.org/web/20171013200331/http:/ifc.recod.ic.unicamp.br/fc.website/index.py?sec=5. [Accessed 17 01 2021].

[75] T. J. d. Carvalho, C. Riess, E. Angelopoulou, H. Pedrini and A. d. R. Rocha, "Exposing Digital Image Forgeries by Illumination Color Classification," *IEEE Transactions on Information Forensics and Security,* vol. 8, no. 7, pp. 1182 - 1194, 2013.

[76] J. Dong, W. Wang and T. Tan, "CASIA Image Tampering Detection Evaluation Database," in *IEEE China Summit and International Conference on Signal and Information Processing*, Beijing, China, 2013.

[77] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data,* vol. 6, p. Article 27, 2019.

[78] W. Wei, J. Li, L. Cao, Y. Ou and J. Chen, "Effective detection of sophisticated online banking fraud on extremely imbalanced data," *World Wide Web,* vol. 16, p. 449–475, 2013.

[79] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature,* vol. 521, p. 436–444, 2015.

[80] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM,* vol. 60, no. 6, p. 84–90, 2017.

[81] M. Buda, A. M. Maciej and A.Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks,* vol. 106, pp. 249-259, 2018.

[82] N. Japkowicz, "The Class Imbalance Problem: Significance and Strategies," in *International Conference on Artificial Intelligence (ICAI)*, 2000.

[83] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng and Y. Kalantidis, "Decoupling Representation and Classifier for Long-Tailed Recognition," in *International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia, 2020.

[84] I. Shrivastava, "Handling Class Imbalance by Introducing Sample Weighting in the Loss Function," Medium, 16 12 2020. [Online]. Available: https://medium.com/gumgum-tech/handling-class-imbalance-by-introducing-sample-weighting-in-the-loss-function-3bdebd8203b4. [Accessed 09 08 2021].

[85] C. Drummond and R. C. Holte, "C4.5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling beats Over-Sampling," *Workshop on learning from imbalanced datasets II,* vol. 11, pp. 1-8, 2003.

[86] N. V. Chawla, N. Japkowicz and A. Kotcz, "Editorial: special issue on learning from imbalanced data sets," *ACM SIGKDD Explorations Newslette,* vol. 6, no. 1, p. 1–6, 2004.

[87] Y. Ho and S. Wookey, "The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling," *IEEE Access,* vol. 8, no. 1, pp. 4806 - 4813, 2019.

[88] Z. Wang, "Practical tips for class imbalance in binary classification," Towars Data Science, 10 08 2018. [Online]. Available: https://towardsdatascience.com/practical-tips-for-class-imbalance-in-binary-classification-6ee29bcdb8a7. [Accessed 09 08 2021].

[89] S. Jadon, "A survey of loss functions for semantic segmentation," in *IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, Via del Mar, Chile, 2020.

[90] R. Vinod, "Dealing with class imbalanced image datasets using the Focal Tversky Loss," Towards Data Science, 07 March 2020. [Online]. Available: https://towardsdatascience.com/dealing-with-class-imbalanced-image-datasets-1cbd17de76b5. [Accessed 30 07 2021].

[91] G. Zhang, X. Huang, S. Z. Li, Y. Wang and X. Wu, "Boosting Local Binary Pattern (LBP)-Based Face Recognition," in *SINOBIOMETRICS: Chinese Conference on Biometric Recognition*, Guangzhou, China, 2004.

[92] A. Alahmadi, M. Hussain, H. Aboalsamh, G. Muhammad, G. Bebis and H. Mathkour, "Passive detection of image forgery using DCT and local binary pattern," *Signal, Image and Video Processing,* vol. 11, p. 81–88, 2017.

[93] D. Cozzolino and L. Verdoliva, "Noiseprint: A CNN-Based Camera Model Fingerprint," *IEEE Transactions on Information Forensics and Security,* vol. 15, pp. 144 - 159, 2019.

[94] K. B. Meena and V. Tyagi, "A Deep Learning based Method for Image Splicing Detection," *Journal of Physics: Conference Series,* vol. 1714, 2021.

[95] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016.

[96] W. Ding, D.-Y. Huang, Z. Chen, X. Yu and W. Lin, "Facial action recognition using very deep networks for highly imbalanced class distribution," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Kuala Lumpur, Malaysia, 2017.

[97] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems,* vol. 25, pp. 1097-1105, 2012.

[98] O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015,* vol. 9351, pp. 234-241, 2015.

[99] A. Chaurasia and E. Culurciello, "LinkNet: Exploiting encoder representations for efficient semantic segmentation," in *Visual Communications and Image Processing (VCIP)*, St. Petersburg, FL, USA, 2017.

[100] H. Zhao, J. Shi, X. Qi, X. Wang and J. Jia, "Pyramid Scene Parsing Network," in *Computer Vision and Pattern Recognition (CVPR)*, Honolulu,, USA, 2017.

[101] X. Zhu, Z. Cheng, S. Wang, X. Chen and G. Lu, "Coronary angiography image segmentation based on PSPNet," *Computer Methods and Programs in Biomedicine,* vol. 200, p. 105897, 2021.

[102] N. J. Francis, N. S. Francis, S. V. Axyonov, S. A. Aljasar, Y. Xu and M. Saqib, "Diagnostic of Cystic Fibrosis in Lung Computer Tomographic Images using Image Annotation and Improved PSPNet Modelling," *Journal of Physics: Conference Series,* vol. 1611, 2020.

[103] P. Banerjee, A. K. Bhunia, A. Bhattacharyya, P. P. Roy and S. Murala, "Local Neighborhood Intensity Pattern–A new texture feature descriptor for image retrieval," *Expert Systems with Applications,* vol. 113, pp. 100-115, 2018.

[104] F. Hakimi, M. Hariri and F. GharehBaghi, "Image splicing forgery detection using local binary pattern and discrete wavelet transform," in *International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, Tehran, Iran, 2015.

[105] Y. Bengio, "Practical Recommendations for Gradient-Based Training of Deep Architectures," in *Neural Networks: Tricks of the Trade*, Springer, 2012, pp. 437-478.

[106] J. Brownlee, "How to Improve Performance With Transfer Learning for Deep Learning Neural Networks," Machine Learning Mastery, 25 08 2020. [Online]. Available: https://machinelearningmastery.com/how-to-improve-performance-with-transfer-learning-for-deep-learning-neural-networks/. [Accessed 05 08 2021].

[107] P. Sutthiwan, Y. Q. Shi, H. Zhao, T.-T. Ng and W. Su, "Markovian rake transform for digital image tampering detection," *Transactions on data hiding and multimedia security,* vol. 6, pp. 1-17, 2011.

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

[109] C. Li, Q. Ma, L. Xiao, M. Li and A. Zhang, "Image splicing detection based on Markov features in QDCT domain," *Neurocomputing,* vol. 228, pp. 29-36, 2017.

[110] Z. Li, K. Kamnitsas and B. Glocker, "Analyzing Overfitting Under Class Imbalance in Neural Networks for Image Segmentation," *IEEE Transactions on Medical Imaging,* vol. 40, no. 3, pp. 1065 - 1077, 2020.

[111] P. Sutthiwan, Y.-Q. Shi, J. Dong, T. Tan and T.-T. Ng, "New developments in color image tampering detection," in *IEEE International Symposium on Circuits and Systems*, Paris, France, 2010.