Image Forgery Detection Using Deep Learning Approach

Siva Sathguru Pandiyarajan

February 2022

School of Mathematics,
Cardiff University

A dissertation submitted in partial fulfilment of the requirements for MSc (in Data Science and Analytics) by taught programme.

CANDIDATE'S ID NUMBER	**2105259**
CANDIDATE'S SURNAME	**Please circle as appropriate** **Mr / Miss / Ms/ Mrs / Rev / Dr / Other… Pandiyarajan**
CANDIDATE'S FULL FORENAMES	**Siva Sathguru**

DECLARATION

This work has not previously been accepted in substance for any degree and is not concurrently submitted in candidature for any degree.

Signed ……Siva Sathguru Pandiyarajan…. (candidate) Date ……07/02/2022……

STATEMENT 1

This dissertation is being submitted in partial fulfilment of the requirements for the degree of ……MSc……(insert MA, MSc,MBA, etc, as appropriate)

Signed ……Siva Sathguru Pandiyarajan…. (candidate) Date ……07/02/2022……

STATEMENT 2

This dissertation is the result of my own independent work/investigation, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A Bibliography is appended.

Signed ……Siva Sathguru Pandiyarajan…. (candidate) Date ……07/02/2022……

STATEMENT 3 –

I hereby give consent for my dissertation, if accepted, to be available for photocopying and for public viewing, and for the title and summary to be made available to outside organisations.

Signed ……Siva Sathguru Pandiyarajan . (candidate) Date ……07/02/2022……

STATEMENT 4 - BAR ON ACCESS APPROVED

I hereby give consent for my dissertation, if accepted, to be available for photocopying and for public viewing **after expiry of a bar on access approved by the Graduate Development Committee.**

Signed …… Siva Sathguru Pandiyarajan ……. (candidate) Date ……07/02/2022……

Executive Summary

Digital photos are the most often used media for visual data transport in the modern era of digitalization. Digital images are now being used in previously unimaginable ways in a variety of industries, including criminal investigation, medical science, journalism, sports, and image forensics, all of which require image authenticity. Images can be altered using a range of tools that are either free or available for a low cost. Certain techniques can distort images to such an extent that the human visual system has difficulty distinguishing between tampered and real images. As a result, detecting image counterfeiting is a difficult task. Over the previous decade, tremendous progress has been made in the field of image forgery detection. However, as image manipulation technologies get more sophisticated, there is still a need for thorough scrutiny in this area.

The purpose of this research is to develop a novel model based on deep learning using error level analysis detection features that are related to entropy and information theory, normalized mutual information in image pre-processing, including a binary cross-entropy loss function in the very last softmax activation layer, and various applications of a label encoder.

Using error level analysis, we can discover image data compression ratios and afterward apply these images to a convolutional neural network to assess whether the image is forged. Experiments demonstrate that by implementing the ELA technique, the CNN model's training efficiency may be significantly increased, and the overall accuracy can surpass 96 percent. In comparison to existing models, the proposed model offers various advantages, including reduced layer count, a shorter training period, and increased efficiency.

Additionally, we covered two key aspects of image forgery detection when utilizing deep convolutional neural networks. We begin by evaluating and comparing a variety of pre-processing techniques, including normalization, error level analysis, and label encoding, which is then fed to convolutional neural network (CNN) architectures. Later, we examined various transfer learning techniques for pre-trained ImageNets (through fine-tuning) and applied them to our CASIA V2.0 dataset. Thus, it discusses pre-processing strategies using a basic CNN model and then examines the transformative effect of transfer learning models. The dissertation also contains proposals for further development of this topic.

Acknowledgment

I would like to express my heartiest gratitude to my professor, Mr. Yukun Lai, who has always guided me in the right direction, and to the University for providing me with this wonderful opportunity to work on this amazing project on the topic of Image Forgery Detection, which also allowed me to conduct extensive research and learn about many new things.

I would like to express my gratitude to my friends and family for all the support during COVID-19 era.

Table of Contents

List of Figures

1. Introduction

As a result of technological advancement and globalization, electronic equipment is now widely and reasonably available, the popularity of digital cameras has increased. There are several camera sensors located throughout our environment, and we use them to acquire a large number of photographs. Photographs are required in the form of a soft copy for various documents that must be filed online, and a great number of images are published on social media each day. The great thing about images is that they can be analysed and information extracted even by persons who are illiterate. As either a result, images have evolved into a key component of the digital world, assisting in the storing and exchange of data. Numerous tools are available for swiftly modifying photos (Xiao, B., 2020). These tools were developed with the express purpose of refining and upgrading photographs. Rather than strengthening the image, however, some individuals use their abilities to fake photos and spread falsehoods (Wu, Y el. June 2019). This is a big issue, as the damage inflicted by fabricated photographs is often severe and irreparable.

There are two fundamental types of image forgery.

- **Image Splicing:** Image splicing is a technique for copying and pasting a portion of an image onto another image. It is frequently followed by undertakings such as blurring, compression, and scaling.
- **Copy-Move:** In this circumstance, there is only one image. Within the image, a portion of the image is copied and pasted. This is a regularly utilised approach for concealing additional items. The resulting forged image is devoid of all prior forged images.

For both forms of forged image, the primary objective is to spread false information by altering the image content. Historically, images were a highly trusted medium for exchanging information; but, due to image counterfeiting, they are now exploited to propagate falsehoods. This undermines public trust in images, as forgeries may or may not be obvious or recognized to the human eye. As a result, detecting image forgeries is critical for preventing the spread of misinformation and re-establishing public confidence in images. All of that is accomplished by examining the numerous abnormalities produced during image forgeries, may be detected to use a variety of image processing methods.

All of the imprints left upon that image even during procurement procedure serve as identifiers for the system is expected and are typically combined to produce created aspects using a myriad of methods. For example, in (F. Marra, 2015) and (A. Tuama, 2016), researchers fed supervised classification algorithm with those pixel descriptors related with co-occurrence statistics. Over

the last decade, convolutional neural networks have grown in popularity in the field of picture forensics. These techniques were developed to teach the CNN the most useful properties for categorizing's evaluations. A benefit of CNN is that it pulls attributes from the image collection directly. While these CNN-based algorithms have the advantage of learning classified features derived via visual data, evaluating the resulting camera identification model is extremely difficult. Another downside of CNN is that the model is data-dependent. Scaling, compression, and filtering of images, on the other hand, are unpredictable. Additionally, fresh image forgeries are being created daily. Thus, researchers must strengthen their suggested model's resilience against all known perturbations.

This study extends previous work (B. Diallo, 2019s), in which a methodology for enhancing the resilience in image forgery detection has been developed. Given that compression is among the most often used types of image editing, we concentrate on strengthening resilience against compression manipulation; nevertheless, a similar technique might be used for any other manipulation. Our framework's initial crucial step is evaluating the clarity of the image data provided for the application under consideration. To accomplish this, we feed the CNN a mixture of compressed and uncompressed images of varied quality. By training the CNN with a variety of image compression qualities, its resilience is strengthened.

Hence, we proposed a very simple CNN well with the primary intention of educating the artifacts that appear together in the tampered image because of feature inconsistencies between both the original and tampered regions. The suggested technique makes the following significant contributions:

- A light CNN-based framework is utilised to detect image fraud efficiently. Unlike the majority of existing algorithms, our technique analyses several artifacts left behind during the image tampering processes and exploits picture source discrepancies via image re-compression.
- Unlike the majority of existing algorithms, our technique analyses several artefacts left there after the image editing process and exploits variances in image sources via image re-compression. Its speed and accuracy are ideally suited for practical implementation, and it is also compatible with slower platforms.

The remainder of the thesis is organised in the following manner. Section two includes background information/a review of the literature on image forgery detection techniques. Section three presents the proposed methods for detecting the presence of forgeries in an image.

Section four explains the methodology and results of the experiment. Finally, Section five summarises the findings, and Section six makes recommendations for the project's future development.

2. Background

Numerous approaches to image forgery have been proposed in the literature. Although the majority of traditional solutions were focused on artifacts leftover from image counterfeiting, techniques neural networks based and deep learning have been introduced more recently, as mentioned below. We will begin by discussing several traditional strategies and then move on to techniques based on deep learning.

2.1 Machine learning in image forgery detection

Machine learning is the use of computer algorithms to improve the learning efficiency of a system. It improves its performance through experience from the operation of a given task and measuring the performance outcome, considering its the task is T performance measured is P and the improvement from experience is E. Machine learning can be primarily divided into supervised and unsupervised learning. Currently, extensive research is being carried out on supervised learning as it has several applications like pattern recognition and classification (Le-Tien et al., 2019). The model developed from supervised learning is based on a dataset of known classes that have been categorized based on sample input data. The advancement of digital transformation, security, and data protection in multimedia and uniqueness has emerged as important questions and challenges in conveying appropriate messages. Multimedia forensics has become a new field of study topic that uses strong machine learning technologies and provides a framework for the development of deep learning to provide accurate information.

Machine and Deep learning can be an excellent way of providing better security and application in forgery detection as image attacks are inherent in these technologies. Several strategies, including deep learning and neural network learning, have been presented in the form of machine learning for securing systems designed for learning outcomes and are emphasized in the detection of image manipulation. The primary aim of these systems is to develop forensic methods based on machine learning by the application of trial-and-error methods (Marra et al., 2020). Support Vector Machine is a system that has been generally used for forensic analyses and developments and modifications are introduced to get a strong foundation selecting the features that can help in training the classifiers. Any technique used for image manipulation

leaves pieces of evidence, even after the most detailed and expert formatting and forensic analysis targets to identify these small but intact pieces of evidence by utilizing processing, coding, and acquisition of the image fragments.

Although deep learning can be applied for digital image forgery detection, there are several limitations in the method that makes it inefficient in the aspects of image security. Dataset training can give excellent results if both "Convolution Neural Network" (CNN) classifiers and support vector machines are applied properly (Yarlagadda et al., 2018). Observational data can be used to teach the computer with the help of "CNN," which is an excellent programming tool. The "CNN" programming model aids the computer in learning from observational data. CNN is a classification of deep neural networks and can help in successfully addressing numerous algorithms that are performed while processing an image, such as pattern recognition; however, it is a complicated computational network due to its interconnection to vast neurons.

The feature extraction function of CNN is an approach driven by data that uses image categorization in the form of shape filters, and for the testing and training process, it requires a vast dataset (Pawan, 2020). The performance of machine learning makes it quite efficient for its application in image security, but it still has several disadvantages, especially in deep learning due to the problem of ensuring the security of large datasets. The test phase in which the output might differ in different trials adds to the list of constraints. Due to these reasons, the necessity to develop a new kind of forensic machine learning process arises (Zhang et al. 2018). As per Machine learning categories, in the classification of different types, two or more classes are assigned to different input types, which can be used by the learner to develop a new model that can classify inputs that have not been detected yet into one or multi-labeled classifications which can then be managed using supervised learning. Next, in the regression system, the problem that was encountered generates continuous outputs that are related and not distinct from each other.

In the clustering system, separate groups are assigned to a set of similar inputs. In clustering, these groups are not known previously, and this is the difference of clustering from classification. This unforeseen grouping makes it a task that cannot be supervised accurately (Zhang et al., 2018). In the dimensionality reduction model, the inputs are categorized according to a lower-dimensional space, thus simplifying their classification. Another type is topic modeling, in which similar topics are identified by analyzing a list of documents consisting of human languages. As per the Classifier Evaluation Process, the entire process is carried out by the learner, and the training of the dataset is based on the preferences of the

learner. Based on his choice, the selection of features and measurements of error is developed by utilizing various strategies for the evaluation of the overall classification model (Nirmala and Thyagharajan, 2019). This process is also driven by data, i.e., the outcomes and results of the model will be entirely based on the data that it is fed on. If the data are appropriate, it might generate good results, whereas undefined and inaccurate data will yield poor results. For example, if a dataset of horse images is fed to the program, then it will generate results for horse images only, and one cannot expect it to derive mouse results from the set. In other words, a model cannot be trained using linear regression if the data does not have a linear correlation.

2.2 Deep learning approach for detection of digital image forgery

With the increasing spread of false images across social media and other online platforms to cause harm and disruption to the public sector, the need for an efficient technique to detect these images is also on the rise so that these attacks can be readily detected and eliminated. Digital image forgery has also been used previously to initiate terror and religious riots in various countries. The general public does not understand these propagandas that are utilized by various communities to cause a misbalance in harmony in the society and fall into the trap of these schemes. For these reasons, it is very crucial to detect forged images so that public harm can be prevented. Today several tools can help in this process. One of them is the detection of the level of compression in an image and measuring the quality of the image. This method is called Error Level Analysis (ELA) (Mohammed et al. 2018). Error Level Analysis, commonly called ELA, is an image analysis technique used in forensic science to determine the various compression levels in an image. This technique is employed to detect images that have been digitally modified. To determine which images have been forged and which are the original version, several strategies are utilized. These strategies are used by researchers often for their study and research purpose.

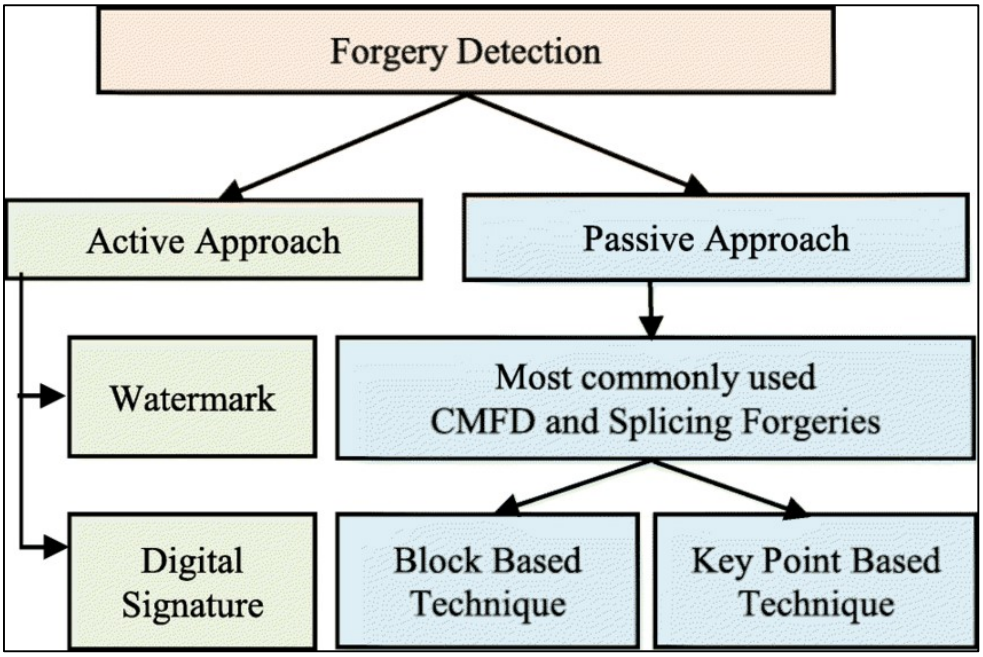


Figure 2.1: Most commonly used techniques to detect image forgery

(Source: Thakur and Jindal, 2021)

One of the strategies has been detailed in the research work, "Forgery Frame Detection From The Video Using Error Level Analysis," by Hites C Patel et al. in which he analyses the frame count of the image and compares them to that of the original image to determine if it is a compressed and fake one (Thakur & Jindal, 2021). Some attributes that are analyzed to identify a compressed file are the number of frames per second, rate of data, rate of the frame, resolution of the image, total bit rate, sample rate of the audio, quality of the video, i.e., what are the pixel size of the video, finding out edited fragments, time length of a video and the Audio Chanel. Meera Mary Isaac and her team utilize the "Gabor Wavelets dan Local Phase Quantization" method to detect image forgery. Birajfar and his team use CASIA TIDE v.1. Dataset is a passive technique to analyze false images. In a study by Youseph et al., researchers shave used the method of illuminated color estimation and combined the HOG edge descriptor along with the canny detection method to identify the edge border of an image. This provided a 74% accuracy value after combining the method with SVM (Sudiatmika and Rahman, 2019). Deep learning is a novel approach that is being applied in the field of machine learning and has been developed recently upon the advent of GPU acceleration. It is a completely new science that has got great potential in different sectors of information and technology. The process of deep learning that has been applied in the detection of false images and identifying forged files is Convolutional Neural Network (CNN) which has been designed to analyse data of two-dimensional forms and is based on the concept of multilayer perceptron (MLP).

2.3 Machine Learning vs. Deep Learning

In deep learning, there are primarily two types of recognition approaches for image forgery detection they are active approach and passive approach. In the active approach, hidden information is extracted from the image. This hidden information can be detected by identifying digital signatures and watermarks (Nirmala and Thyagharajan, 2019). In the passive approach, region duplication is identified in the image. These duplications could have been generated by the use of splicing or CMF on a given image. The difference between CMF and SF is that, while a few patches of an image are replaced by a modified fragment of the same image in CMF, a few fragments are obtained from an image and pasted on another image to modify the latter (Bouktif et al. 2018). These are the most commonly used forms of forgery, and to detect modifications done using these techniques, methods based on key-point and blocks are used. In a block-based method, the images are divided into several overlapping blocks that lead to complexity in the time frame, thus helping to compare the time frame of the original image with a fake one. Key point-based methods are able to identify a few key points from an image, and in other cases; it is unable to identify the forgeries.

KNN, in combination with SIFT algorithm, is used in machine learning to identify image forgery and detect modified fragments, whereas, SVM classifies original pixels from forged ones and performs localization. The primary difference between machine learning and deep learning is that Machine learning is a subcategory of Artificial Intelligence, whereas, Deep learning is a subclass of Machine Learning. According to machine learning, a system improves and learns from previous experiences just like a human. The performance of a system can be improved from the experiences that the system has gained from the tasks it has performed previously (Liu and Lang, 2019). Sundar Pichai, CEO of the IT giant Google, once stated in a 2017 Google event that computers have gained a superior capability of image recognition and are better in this aspect than humans. It has been almost 80 years since Machine learning has been used for detecting image forgery but what plays the most significant role in the advancements of machine learning are the improvements in the GPU. These improvements are the pioneers who are providing the next for machine learning processes (Liu and Lang, 2019). Two recent trends are primarily used by Machine learning and deep learning, in which, one is large datasets accessible for the rigorous training of algorithms, and the other is advancements of GPU technology.

2.5 Associated works to the research

T. J. de Carvalho et al. (T. J. de Carvalho, 2013) advanced a machine learning-based detection approach. Splicing forgery is a sort of image forgery in which different parts from multiple

photos are collectively pasted and a new image is created. It took advantage of the dissimilarity idea in colour illumination. The extraction of characteristics was aided by the SVM (state-vector machine) classifier, a machine learning algorithm. The results indicated an accuracy of 86 percent for internet photos and cross-database training/testing.

Z. J. Barad and M. M. Goswami (Z. J. Barad and M. M. Goswami,2020) summarise the analysis and findings in a way that other researchers can understand. It summarised many research studies and their findings. Provided details about the datasets used to identify image fraud. It discusses two prominent techniques for detecting forgeries: (a) traditional (b) deep learning (DL). They were concluded by claiming that deep learning algorithms outperform traditional approaches by a wide margin. As deep learning is a two-stage process that begins with feature extraction and ends with classification. These methods perform admirably on even the most complex feature datasets. The paper discussed the classification of strategies for detecting image forgery. To investigate the various forms of deep learning networks, a comparison of Deep Neural Networks, Recurrent Neural Networks, and Convolutional Neural Networks is undertaken.

In (Fahime Hakimi *et al.*, 2015), the author provides a novel strategy for detecting frauds of image splicing that is composed of three methods: an Support Vector Machine(SVM) classifier, an Local Binary Pattern(LBP), and a Principal Component Analysis(PCA). The creator of this technique transformed the input image RGB to the YCbCr colour channel first and then formulated the chrominance component into nonoverlapping blocks. Second, the Local Binary Pattern operator is used, followed by the wavelet transform in all blocks. Finally, all blocks are subjected to Principal Component Analysis (PCA) and the resulting features are supplied to the Support Vector Machine (SVM) classifier. The approach was evaluated in practice using two different research datasets, CASIA and Columbia.

On the other hand, typical deep learning frameworks must not be utilised directly since modified images were difficult to distinguish from originals when a variety of existing image modification methods are applied, which require alteration of the input and architecture. (2016, Y. Zhang). ELA is a technique that may be used to determine whether or not an image has been altered. This technique detects faults by downgrading the image's quality and then estimating the error level on an 8x8 grid. Whereas if the image is not changed, the error level will be the same for all eight grids. (2017) (T. S. Gunawan). To detect digital image forgeries, a combination of ELA characteristics and CNN architecture can be used (T. S. Gunawan, July 2017). However, no research on the efficacy of ELA in this endeavor has been undertaken. As

such, the goal of this work is to illustrate how incorporating ELA properties affects the effectiveness of CNN-based picture detecting fraud results.

2.6 Future directions

Today there are several techniques available from machine learning and deep learning that can be used to detect digital image forgery; however, the need to keep improving these existing systems is important. A feature that can automatically extract pieces of information from forged images to ease the process of detection for the system is required as images that are fake and have been forged are rising day by day and are being used to threaten society. This creates a vast scope for forgery detection systems using machine learning and deep learning. Better techniques to detect forgery that has been accomplished using splicing and hybrid copy-move as well as classification of forged fragments using "semantic segmentation", "deep convolution neural network" and color illumination is a possible area of improvement (Meena and Tyagi, 2020). The application of these techniques is especially important in the fields of news, banking, financial services, education, healthcare, and online shopping (Ghoneim et al., 2018). These sectors are susceptible to fraudulent images and can cause damage to the human resources of companies.

The state-of-the-art(Ali et al., 2022) approaches detecting image manipulation (Ali et al., 2022) frequently take an inordinate amount of time to process. They can identify both image splicing or copy-move forgeries, never both. Some other key concern with them is their lower degree of accuracy in detecting forgeries. As a result, a more efficient and precise framework is required. We addressed this issue by introducing a novel image recompression-based approach (Error level analysis). Apart from increasing the accuracy of picture forgery detection, our suggested system also improves response time.

3. Research Methodology

This section discusses how to detect faked photos. To get started, we'll download the dataset (CASIA V2.0). The data sets are then normalized and the ELA approach is used to highlight regions of the original image with an error level higher than a threshold value, indicating that the affine translation created artifacts. Our research is divided into two phases: one in which pre-processing, and CNN architecture are compared with ELA, without ELA, and with ELA-Enhance. Finally, we performed a comparative analysis using two pre-trained models,

MobileNetV2 and ResetNet50 (fine-tuning), to obtain much better results. We trained our dataset using two distinct transfer learning techniques and selected the best one.

3.1 Data Collection Method

Image forensics has heightened public concern about justice, as increasing instances of altered photos being used as evidence in media and in court have been recorded recently. To ensure the validity of image content, passive-blind image tampering identification is required. Additionally, more accurate open benchmark databases were required to aid in the techniques' development. We just compiled a library of natural color images subjected to realistic manipulation techniques. The database is made freely available to researchers for the purpose of comparing and evaluating their suggested tamper-detection algorithms which is known as the CASIA Image Tampering Detection and Evaluation Database.

CASIA ITDE database V2.0 has a similar structure to database V1.0, but it is an extended version. It contains 12,323 color photos in total, as well as two image subsets (authentic and tampered). There are 7,200 legitimate photographs in the authentic set and 5,123 altered images in the tampered set. The images in V2.0 have a variety of sizes, range between 800 x 600 pixels to 1600 x 1200 pixels. V2.0 includes both uncompressed image examples (BMP and TIFF) and JPEG images with variable Q factors. The real images in V2.0 were gathered from the Corel image database [Corel Database], publicly accessible websites, and our own acquired images.

Dataset	Authentic Images	Tampered Images	Total Images
CASIA.2.0	7491	5123	12,614
Training Dataset (80%)	5993	4098	10,091
Testing Dataset (20%)	1498	1025	2523

Figure 3.1 CasiaV2 dataset Split-up

3.2 Pre-processing Method

The following are the pre-processing steps that are used in many comparative studies of CNN architecture and deep learning approaches in order to obtain significantly improved results.

3.2.1 Image Normalization

In image processing, normalization is a method that adjusts the range of pixel intensity values. Examples include low-contrast images caused by glare. Occasionally, the term "normalization" is used synonymously with "contrast stretching" or "histogram stretching." In more general

data processing fields, such as digital signal processing, this is known as dynamic range expansion.

In most cases, the purpose of dynamic range expansion is to put an image or other type of signal together into a range that is more known or regular towards the sense, hence the term normalization. The motivation is often to preserve the image contrast of the data collection, signals, or images in order to prevent mental distraction or tiredness. As an example, a newspaper will make an effort to ensure that all images inside an issue have the same grayscale range.

Normalization converts an n-dimensional grayscale image to a two-dimensional image.

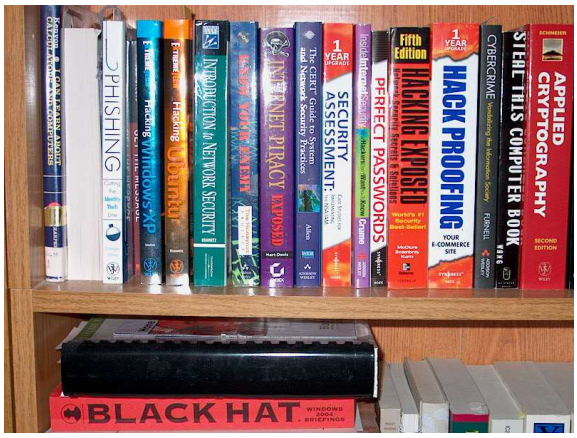
The linear normalisation formula for a grayscale digital image is

$$I_N = (I - \text{Min})\frac{\text{newMax} - \text{newMin}}{\text{Max} - \text{Min}} + \text{newMin}$$

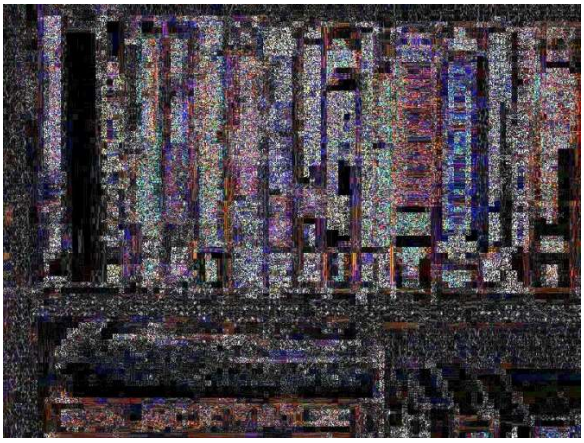
3.2.2 Error level analysis

An error level analysis (ELA) technique is one method of determining whether or not photographs have been altered. It accomplishes this by saving photos at a specified level of quality and then calculates the difference between the quality level and the compression level (Jeronymo DC,2017). When JPEG was saved for the first time, it shrinks the image; JPEG compression is supported by the majority of editing programs, including Adobe Photoshop, Gimp, and Adobe Lightroom. The image must first be compressed before it can be scheduled with image editing tools. Thus, when the first image is shot with a digital camera, the original image is compressed twice, once by the camera and once by editing software. The image seems same when viewed with the naked eye; however, when viewed through this means, it is possible to tell the difference between a fake and the actual image. Calculation of the average difference between Y (luminance) and CrCb quantization tables (Chrominance). The image is not optimized for a particular camera quality level by the digital camera (high, medium, low, etc.). Original digital camera photographs should have a high ELA value. Each consecutive resave reduces the possibility of an error. As seen in Figure 2, original images have a high ELA value, which is visible in the ELA image as white. There is no obvious difference when the image is resaved using standard human vision, but ELA displays mostly black and dark colors. If this image is resaved, the quality of the image will deteriorate. If the input image is then altered, the affected area will display color with a higher ELA level.

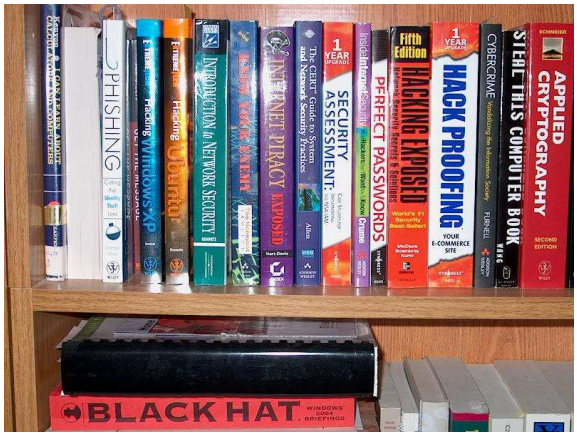
Figure 2 illustrates the effect of ELA's output on the image's condition.



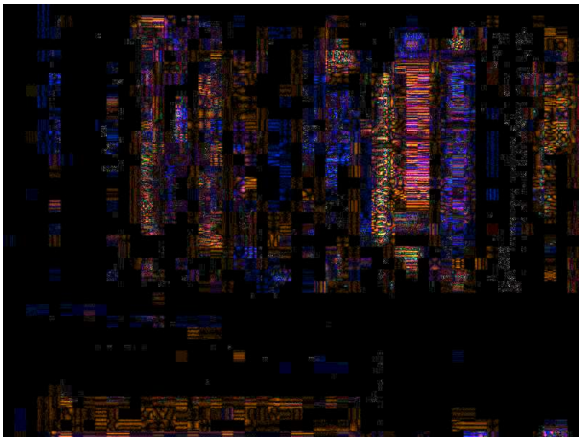
(a) original image



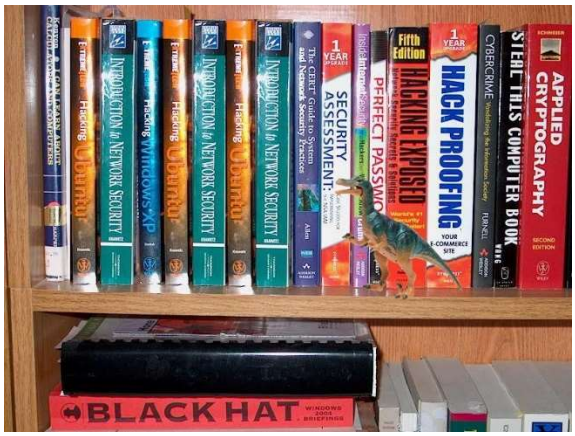
(b) ELA original Image



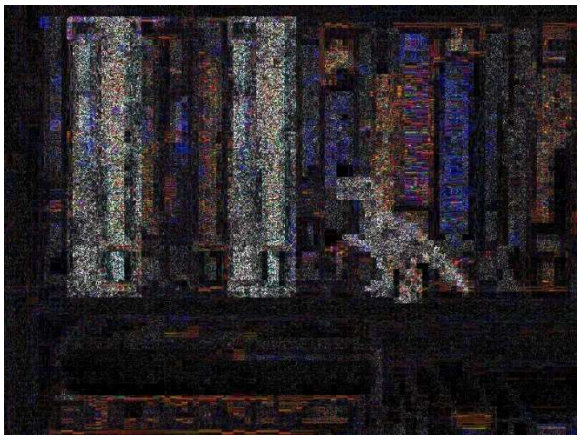
(c) resave image



(d) ELA resave image



(e) tampered image



(f) ELA tampered images

Figure 3.2 Error level analysis compression

(Source: FotoForensics. *et al.* 2020)

3.2.3 Label Encoding

We will substitute the categorical value with a numerical value between 0 and 1 during label encoding. Label 0 indicates tampering, while label 1 indicates authenticity.

3.3 Deep learning Architecture

CNNs are a type of non-linear interconnected neurons inspired by the visual system of the human. They have already demonstrated extraordinary potential in a variety of computer vision techniques, such as image segmentation and object detection. They may also be advantageous for a variety of other reasons, such as image forensics (Ali et al., 2022). Because image fraud is incredibly disruptive and detection is crucial, image forgery is relatively straightforward to conduct with the several tools available today. Whenever a segment of such an image is transported from one area to another, a variety of artifacts form due to the images' disparate origins. Since these artifacts are typically imperceptible to the naked eye, CNNs may detect them in created images. Due to the fact that the origin of the forged region and the background images are dissimilar when such images are recompressed, the forged region is enhanced differently due to the compression difference. In the suggested strategy, we apply this principle to train the CNN model and analyse whether an image is genuine or a forgery.

The below figure shows the architecture of CNN model.

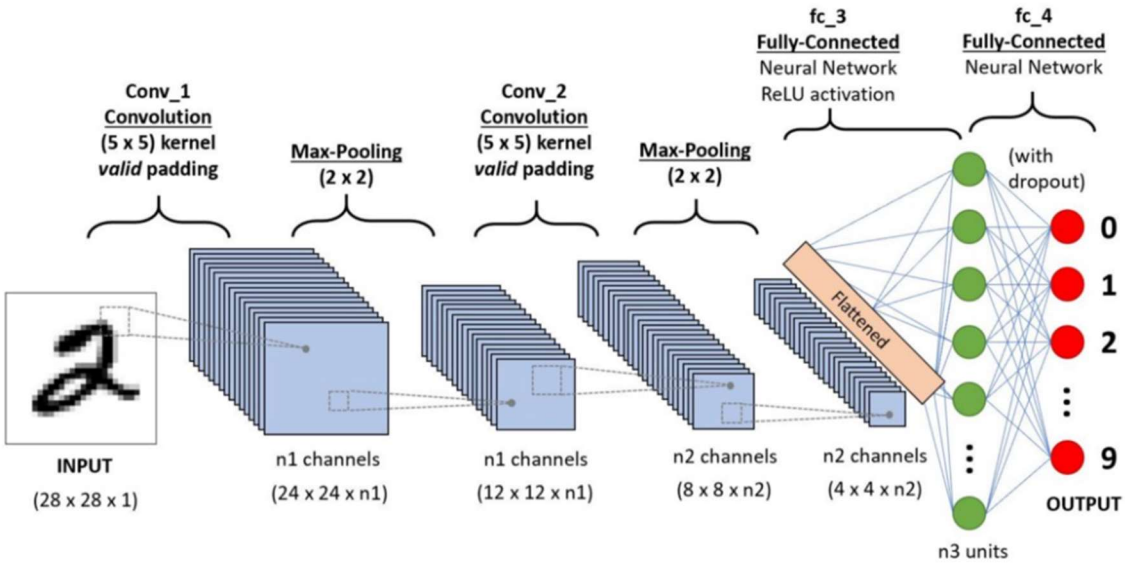


Figure 3.3 CNN Architecture

3.4 Transfer learning Architecture

Transfer learning is a technique that enables the final layer of an existing architecture to be retrained, hence reducing training time with a small dataset.

3.4.1 ResNet50

ResNet50 is a variant of ResNet with 48 Convolutional layers, 1 MaxPool layer, and 1 Average Pool layer. It supports floating-point operations up to 3.8 x 10^9. It is a regularly used ResNet model, and we have conducted substantial research on the ResNet50 architecture.
The architecture is shown in below figure.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 3.4: The ResNet 50 architecture

As illustrated in Figure 4, ResNet 50 architecture includes the following component:

- A convolution with a kernel size of 7 * 7 and 64 distinct kernels, each with a stride size of 2, results in a single layer.
- Following that, we see maximum pooling with a stride size of two.
- Following there comes a 1 * 1,64 kernel, followed by a 3 * 3,64 kernel, and finally a 1 * 1,256 kernel. These three layers are repeated three times in total, giving us nine levels in this stage.
- Following that, we see a kernel of 1 * 1,128 followed by a kernel of 3 * 3,128 and finally a kernel of 1 * 1,512. This phase was done four times, totalling 12 layers.
- Following that, there is a 1 * 1,256 kernel and two further kernels with 3 * 3,256 and 1 * 1,1024, which are repeated six times for a total of 18 layers.
- And then again, a 1 * 1,512 kernel with two more of 3 * 3,512 and 1 * 1,2048 and this was repeated 3 times giving us a total of 9 layers.
- Following that, we perform an average pool and conclude with a fully linked layer having 1000 nodes and a softmax function, which results in a single layer.
- We omit the activation functions and the max/average pooling layers from our count.

Thus, when we add these together, we get (1 + 9 + 12 + 18 + 9 + 1 = 50) layer for Deep Convolutional network. I have choosed ResNet 50 for my thesis comparison research due to its performance on the ImageNet validation set.

3.4.1 MobileNetV2

MobileNetV2 is a convolutional neural network design optimised for mobile device performance. It is built on an inverted residual structure in which the remaining connections between the bottleneck layers are the bottleneck layers themselves. The intermediate expansion layer filters features using lightweight depth-wise convolutions as a source of non-linearity. MobileNetV2's architecture as a whole begins with a fully convolutional layer of 32 filters, followed by 19 residual bottleneck layers.

We have thoroughly examined the MobileNet V2 architecture. The MobileNet V2 model contains 53 convolution layers and 1 AvgPool, resulting in a total of almost 350 GFLOP. It is composed of two major components:

- Inverted Residual Block
- Bottleneck Residual Block

Convolution layers are classified into two categories in the MobileNet V2 architecture:

- 1x1 Convolution
- 3x3 Depth-wise Convolution

The overall architecture is as shown in below Figure.

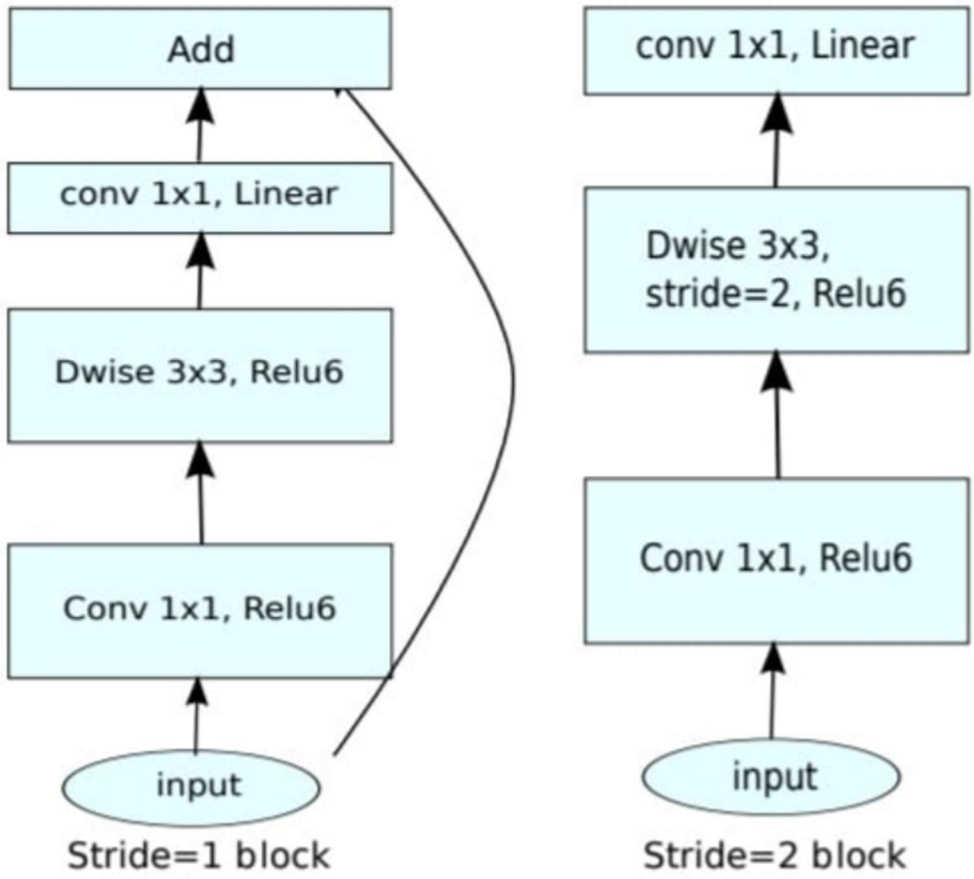


Figure 3.5: The MobileNetV2 architecture

MobileNetV2 supports two distinct types of blocks. The first is a recurring block of one stride. A block with a stride of two is another option for decreasing.

3.5 Critical Factors – developing models

Early stopping

Overfitting is a critical factor to take into account while training a neural network with sample data. Whenever the number of epochs used to train a neural network model exceeds the required number, the training model rapidly learns patterns specific to the sample data. This makes it impossible for the model to perform effectively on a new dataset. This model performs admirably on the training set (sample data), but not on the test set. As a result, we've implemented early halting to monitor this.

Optimizer

Also, I have focused on seeing if the choice of optimizers and learning rate could have a significant impact on the performance of CNN models. The learning rates tested are 0.0005, 0.0001, 0.005, 0.001, 0.05, 0.01, 0.1, which have been implemented on the customized CNN model and pre-trained Resnet50 as it has better accuracy compared to the other model. Comparative study of various optimisers has been done and shown with the results in the below section.

Epochs

We examined how the validation loss behaves following each epoch in this experimental study. If the loss reaches a saturation point, we have reached the desired number of epochs. However, if the validation loss increases, this is a sign of overfitting, which we have addressed by incorporating certain regularisation strategies (like data augmentation, batch normalisation, dropout and early stopping).

Batch_size

The batch size specifies the number of samples produced prior to updating the model.

4. Design and Implementation

The first stage of the implementation process is data pre-processing where the whole dataset is resized. It makes the entire dataset in a normalized format. After that, modelling is performed for identifying fraudulent images from the given image dataset. In this case, sequential modelling is used (Ma et al. 2022). After that, learning of the model is transferred through ImageNet and Confusion Matrix.

4.1 Preliminary Setup

We evaluated the suggested technique's efficiency using a CASIA 2.0 dataset. There are 12,614 images of which 7491 are authentic and 5123 are tampered images. CASIA 2.0 contains images in the BMP, JPG, and TIF formats. The images in the database come in a variety of sizes; the resolution of the images vary between 800x600 and 384x256 pixels. For this experiment, a web-based application is designed to interact with the neural network model. For creating the user interface, colab is used which is hosted Jupyter notebook service. It is useful for writing and executing Python library through a web browser (Mandal et al. 2022). It is well suited for machine learning related applications. It executes in cloud framework. The web-based application provides following features to the users.

- It performs a detailed analysis of the images to distinguish whether these are real or fake.
- It provides a graphical overview about model accuracy and model loss.
- It features a single 12GB NVIDIA Tesla K80 GPU that may be operated continuously for up to 12 hours.

In this case, computational power of Python programming can be used to create a highly interactive web-based application for detecting fraudulent images from the given image data set (Mandal et al. 2022). This application can be deployed in any servers depending upon the requirements of the end users.

4.2 Proposed Method 1

The initial step was to split the CasiaV2.0 dataset into two categories: authentic and fabricated images. By converting the image to an image of 128x128 pixels, we normalize it. Then our next step is to perform error level analysis. In this method, we are making a comparison between CNN, CNN with ELA, and CNN with ELA_ENHAN. The datasets are split into training and testing with the ratio of 80 percent and 20 percent respectively, which is then passed to the models as shown in below flow chart.

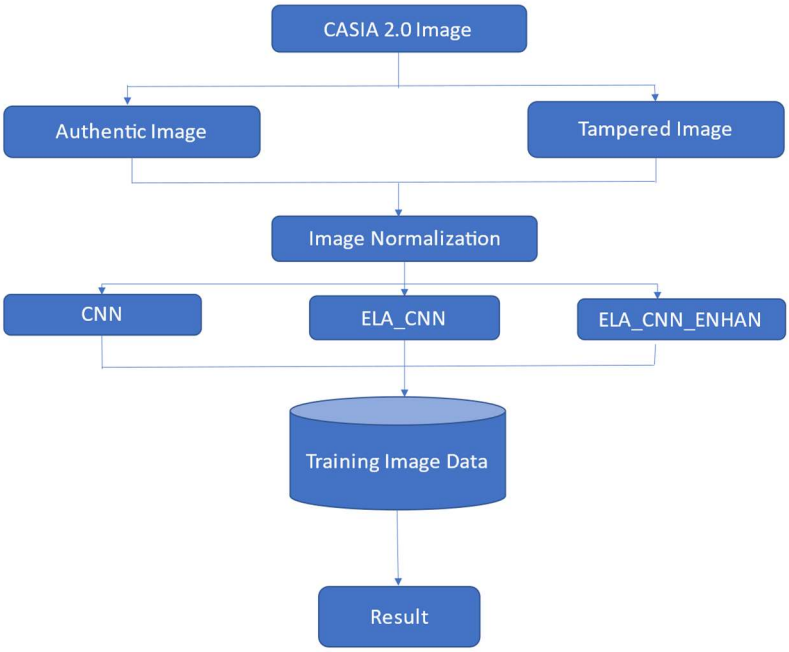


Figure 4.1: Method 1 flowchart

We use ELA to process and obtain the ELA images for the 128 x128 ROI region images. Rather than using the actual images, we trained a CNN model using these ELA images.
Converting the original image to an ELA image improves the efficiency of the CNN model's training. Due to the fact that the ELA image contains less information than the original image, the efficiency can be increased. The ELA image feature concentrates on the region of the original image with an error level greater than the threshold value. Additionally, because the pixels in the ELA image are frequently highly unlike the pixels nearby, and although the contrast is quite visible, the image processed by ELA improves the effectiveness of the CNN training model. To further improve the model's performance in detecting image forgery, we tested the combination of ELA and a sharpen filter. Both pre-processing procedures are important in this situation, as evidenced by the results. Sharpening is the act of increasing the contrast between bright and dark pixels in order to bring out features more clearly (Ding et al., 2018),(Cao et al., 2011),(Cao et al., 2009).

The sharpen filter is employed to create the brighten effect; the pixels are increased in relation to their surroundings. By applying this filter to the edited image, the contrast process will be uneven, as the modified(tampered) image's edges and lines have been blurred or warped.

As a result, we train a CNN model to extract characteristics from the ELA photos and then determine whether or not the input image has been tampered with. Only two convolution layers are necessary for the architecture we adopt because the ELA pictures generated during the conversion process might highlight characteristics of the original image with an error level greater than the threshold value. As a result, determining whether an image is genuine is simplified.
The dataset is divided into training and testing segments with a ratio of 80% and 20%, respectively, and then passed through the CNN and ELA_CNN combination.

Algorithm for ELA feature extraction:

for $epochs = 1$ to $total\_epochs$ **do**
  $training\_error = 0$
  for $i = 1$ to n **do**
    $A_{recompressed\_i} = JPEG_{Compression}(A_i, Q)$
    $A_{diff\_i} = A_i - A_{recompressed\_i}$
    $A_{reshaped\_diff\_i} = reshape(A_{diff\_i}, (128, 128, 3))$
    $training\_error = (L_i - \text{Image\_Forgery\_Predictor\_Model}(A_{reshaped\_diff\_i})) + training\_error$
  end for
  modify_model($training\_error$, Image_Forgery_Predictor_Model(), Adam_optimizer)
end for

The next step would be to predict if a image is forged or not, hence we label the images using label encoding as per below:

Algorithm for Label encoding:

/* *Image forgery prediction (line 25 to 32)* */
Input: Image '*Input_Image*'
Output: '*Input_Image*' labelled as tampered or untampered
$Input\_Image_{recompressed} = JPEG_{Compression}(Input\_Image, Q)$
$Input\_Image_{diff} = Input\_Image - Input\_Image_{recompressed}$
$Input\_Image_{reshaped\_diff} = reshape(Input\_Image_{diff}, (128, 128, 3))$
$Predicted\_label = \text{Image\_Forgery\_Predictor\_Model}(Input\_Image_{reshaped\_diff})$
/* If *Predicted_label* [0][0]>*Predicted_label* [0][1], then *Input_Image* is tampered
/* If *Predicted_label* [0][1]>*Predicted_label* [0][0], then *Input_Image* is untampered

Algorithm for CNN model:

/* Prediction Model Description */

Forgery_Detection(image with size 128 × 128 × 3)

{

convolution Layer1: 32 filters (size 5 × 5, strid=1, activation: “relu”)

convolution Layer2: 32 filters (size 5 × 5, strid = 1, activation: “relu”)

convolution Layer3: 32 filters (size 5 × 5, strid = 1, activation: “relu”)

Max-pooling of size 2 × 2

Dense layer of 256 neurons with “relu” activation function

Two neurons (output neurons) with “sigmoid” activation

}

Model Description

In our proposed model, we created a very lightweight CNN model with few parameters. As described below, we created a model consisting of three convolutional layers followed by a dense, fully connected layer:

- The first convolutional layer consists of 32 filters of size 5-by-5, stride size one, and "relu" activation function with an input shape of 128x128 and a dropout value of 0.1.
- The first convolutional layer is composed of 32 filters of size 5-by-5, stride size one, and "relu" activation function with an input shape of 128x128 and a dropout value of 0.1.
- The second convolutional layer is composed of 32 filters of size 5-by-5, stride size one, and "relu" activation function with a dropout value of 0.2.
- Finally, we have the dense layer composed of 256 neurons with "relu" activation function, which is connected to two neurons (output neurons) with “sigmoid” activation.

4.3 Proposed Method 2

The second method for additional analysis is active learning, also known as transfer learning (Shin et al., 2016), (Chelghoum R et al., 2020), (Shaha et al., 2018), (Hussain et al., 2018). Transfer learning is a design methodology in which we take knowledge from one task and apply it to improve the performance of another task.

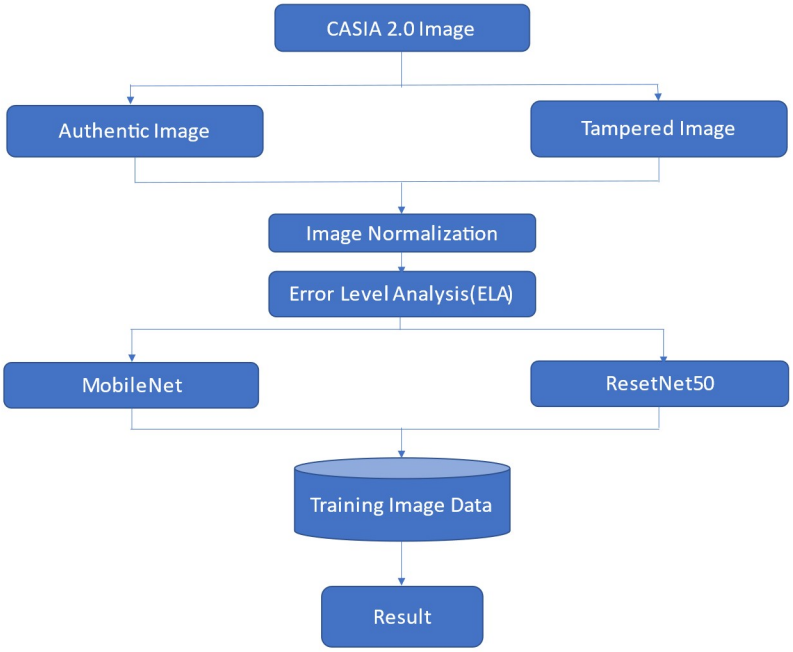


Figure 4.2: Method 2 flowchart

Transfer learning is used in this study to leverage ResNet50's experience for image forgery detection. Two stages comprise transfer learning: freezing and fine-tuning.

At the freezing stage, the pretrained model's publicly available weights and learning parameters are frozen and used "as is." The fine-tuning process begins by removing the ResNet50's fully connected layer (FC) and reconstructing it as three fully connected layers, each with two output neurons corresponding to the forged and authentic image. Take note that the FC layer's weights are initialised randomly during training. The remaining layers' weights are frozen to ensure that they operate as a strong feature extractor for images with a high level of abstraction, as they have previously been trained on millions of images from the ImageNet dataset. The network is trained and tested with ELA generated images a, batch size consisting of 32 images per iteration.The initial learning rate and reducing factor of the fully linked layers are set at 0.0001 and 0.1, respectively, to minimise the cost function during training. Finally, we can combine these into a sequential model and optimise it using a Binary Crossentropy loss and ADAM. The choice of epochs is a challenging problem because it is closely related to the number of optimizations performed during training. With a large number of epochs, the network may get overfit and perform poorly. We monitored the error and performance rates on validation images to avoid overfitting. At epoch 11, ResNet50 had the highest training accuracy and generalisation capability.

Arguments for transfer learning models.

ResNet50

input shape: Image shape is set as (128,128,3) as the include top is False.

alpha: Defines the network's width. By default, this value is set to 1.0.
depth multiplier: Multiplier for the depth component of depthwise convolution. By default, this value is set to 1.0D
dropout: Dropout rate. We utilised a dropout value for 3 of the dense layer as 0.1, 0.2 and 0.3.
include top: Boolean, whether to include the fully-connected layer at the top of the network. set to false as the output of the preceding layer, and passed via a GlobalAveragePooling2D() function that extracts the most significant features.
weights: The weight is set to 'imagenet' (pre-training on ImageNet).
pooling: average pooling is chosen so that global average pooling is applied to the output of the last convolutional block, resulting in a 2D tensor as the output of the model.
Classes: An optional number of classes into which to classify images; required only if include top is True and no weights argument is supplied. By default, 1000 is used.
classifier activation: In the last dense layer, softmax is chosen as the activation.

Epoch: 15
Optimiser: ADAM
Learning rate: 0.001
Batch size:32

MobileNetV2

The MobileNetV2 design is based on an inverted residual structure, with the residual blocks' input and output being thin bottleneck layers. Additionally, it employs lightweight convolutions to filter the expansion layer's characteristics. Finally, non-linearities in the narrow layers are eliminated. We are instantiating a MobileNetV2 model with classification layers that are dependent on the final layer before the flatten operation. This can be altered with the include top argument, but is often ineffective due to the loss of generality in images when compared to bottom layers. Following that, we freeze the convolutional layers and utilise the base model to extract features. We will need to convert the feature vectors to predictive models. This can be accomplished by applying a Global Average Pooling 2D layer that converts the feature vector to a 1280 element vector. Then, using a Dense layer, we may achieve the final prediction. Finally, we can combine them into a sequential model and optimise it using a Binary Crossentropy loss and the RMSProp optimizer.

Arguments implemented are:

input shape: Image shape is set as (128,128,3) as the include top is False.

alpha: Defines the network's width. By default, this value is set to 1.0.

depth multiplier: Multiplier for the depth component of depthwise convolution. By default, this value is set to 1.0D

dropout: Dropout rate. We utilised a dropout value of 0.3 in the dense layer.

include top: Boolean, whether to include the fully-connected layer at the top of the network. set to false as the output of the preceding layer, and passed via a GlobalAveragePooling2D() function that extracts the most significant features.

weights: The weight is set to 'imagenet' (pre-training on ImageNet).

pooling: average pooling is chosen so that global average pooling is applied to the output of the last convolutional block, resulting in a 2D tensor as the output of the model.

Classes: An optional number of classes into which to classify images; required only if include top is True and no weights argument is supplied. By default, 1000 is used.

classifier activation: In the last dense layer, softmax is chosen as the activation.

Epoch: 15

Optimiser : RMSprop

Learning rate: 0.001

Batch size:32

Early stopping : enabled over validation loss

5. Results and discussion

The results of experiments are provided in this section in detailed format.

5.1 Effect of Optimisers with different learning rate for the best-chosen model

We chose the best models from the two approaches (ELA_CNN_Enhan and ResNet50) and evaluated them by adjusting the optimisers (ADAM, RMSprop, and SGD) and the learning rate from 0.0001 to 0.1. This analysis was extremely beneficial in comprehending the effect of the learning rate and optimisers on not only accuracy improvement but also on estimating the loss function. With the result from the models, we can see how much each optimiser is working towards minimizing the loss function for the set learning rates. We have seen that the Adam with 0.001 learning has the highest accuracy and lowest loss function.

Optimisers		Average Training Accuracy		Average Validation Accuracy		Average Training Loss		Average Validation Loss	
	Learning Rate	Customized CNN	ResNet50	Customized CNN	ResNet50	Customized CNN	ResNet50	Customized CNN	ResNet50
ADAM	0.0001	0.92	0.91	0.91	0.87	0.17	0.23	0.21	0.34
	0.0005	0.97	0.97	0.92	0.89	0.05	0.06	0.19	0.23
	0.001	**0.97**	**0.96**	**0.94**	**0.95**	**0.07**	**0.08**	**0.13**	**0.1**
	0.005	0.88	0.86	0.84	0.83	0.25	0.31	0.31	0.36
	0.01	0.91	0.89	0.88	0.87	0.18	0.21	0.25	0.27
	0.05	0.78	0.81	0.78	0.79	0.52	0.52	0.51	0.47
RMSprop	0.1	0.78	0.78	0.78	0.78	0.52	0.52	0.51	0.51
	0.0001	0.86	0.86	0.6	0.6	0.27	0.27	0.56	0.56
	0.0005	0.92	0.92	0.9	0.9	0.18	0.18	0.21	0.21
	0.001	**0.97**	**0.97**	**0.92**	**0.92**	**0.06**	**0.06**	**0.19**	**0.19**
	0.005	0.94	0.94	0.91	0.91	0.18	0.18	0.22	0.22
	0.01	0.93	0.89	0.9	0.87	0.19	0.21	0.22	0.27
	0.05	0.77	0.81	0.78	0.79	0.52	0.52	0.51	0.47
SGD	0.1	0.77	0.78	0.78	0.78	0.52	0.52	0.49	0.51
	0.0001	0.78	0.86	0.78	0.6	0.63	0.27	0.62	0.56
	0.0005	0.78	0.78	0.78	0.78	0.51	0.51	0.5	0.5
	0.001	0.78	0.86	0.78	0.83	0.51	0.31	0.5	0.36
	0.005	0.78	**0.89**	0.78	**0.87**	0.42	**0.21**	0.39	**0.27**
	0.01	0.85	0.81	0.87	0.79	0.31	0.52	0.29	0.47
	0.05	**0.86**	0.78	**0.88**	0.78	**0.28**	0.52	**0.26**	0.51

Figure 5.1: Optimiser comparative study

5.2 Experiments and results

Method 1 outcome:

MODEL	Train Accuracy (%)	Validation Accuracy (%)	Train Loss (%)	Validation Loss(%)	Number of Epocs
CNN	74.29	62.15	50.12	65.56	8 (Early stopping)
CNN-ELA	93.19	92.73	16.42	17.24	30
ELA_CNN_Enhanc	97.09	94.43	7.23	13.61	30

Figure 5.2: Method 1 outcome

In Method 1, we conducted a comparative study between a CNN with ELA, a CNN without ELA, and a CNN with ELA sharpening. As shown in the above graph, the CNN with sharpen filter has significantly higher accuracy and minimum loss of 97 percent and 7%, respectively, resulting in an improved margin of 19.94 percent and 47 percent compared to the previous method.

The confusion matrix below summarises the outcomes of a classification problem's prediction. The number of correct and incorrect predictions is summed and classified using count values. We were effective in predicting 1429 authentic photos and 350 manipulated images using the Customised CNN model.

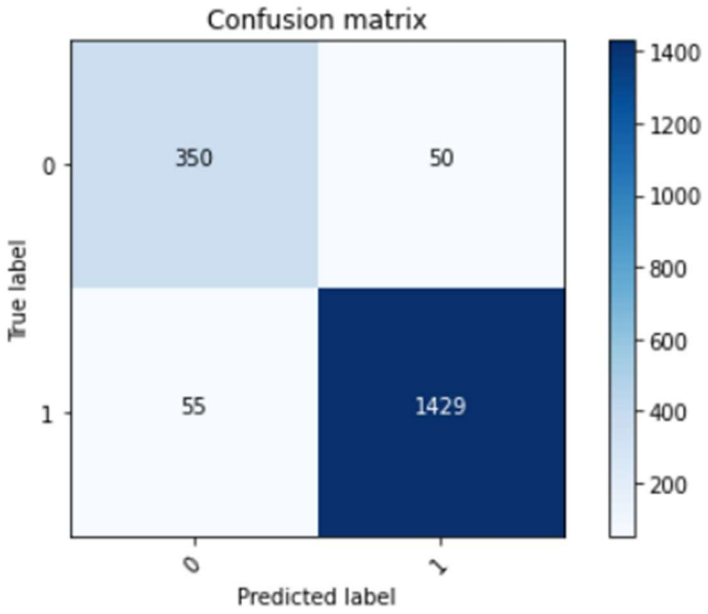


Figure 5.3: CNN-ENHAN confusion matrix

Method 2 Outcome:

MODEL	Train Accuracy (%)	Validation Accuracy (%)	Train Loss (%)	Validation Loss(%)	Number of Epocs
ResNet50	96.73	95.54	8.28	10.73	15
MobileNetV2	99.34	80.25	2.5	74.91	15

Figure 5.4: Method 2 outcome

We chose two CNN transfer learning light frameworks for this strategy, Resnet50 and MobilenetV2, and found that ResNet50 is significantly more accurate than MobilenetV2 in terms of confusion matrices and average accuracies.

Additionally, we can observe that validation loss for MobileNetV2 is increasing, resulting in model overfitting. We attempted to correct various adjustments, such as image augmentation, dropout, batch normalisation, and early stopping, but were unable to do so. As a result, this will be considered as one of the future areas of improvement.

Confusion matrix for RestNet50:

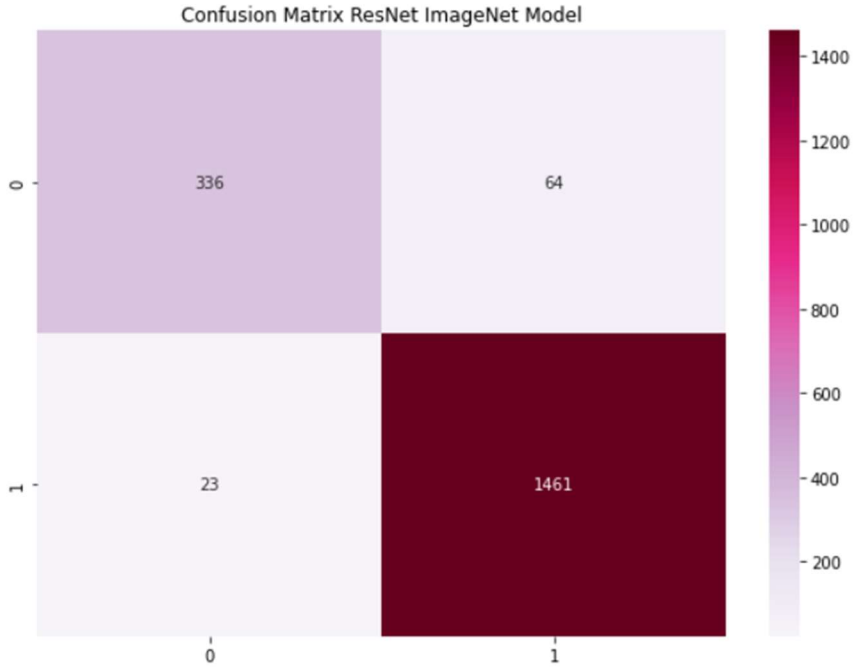


Figure 5.5: ResNet50 confusion matrix

Also we have plotted the graphs for accuracy vs loss as per the above discussed methods:

Graphs for Method 1 (CNN with ELA_ENHAN) is shown below:

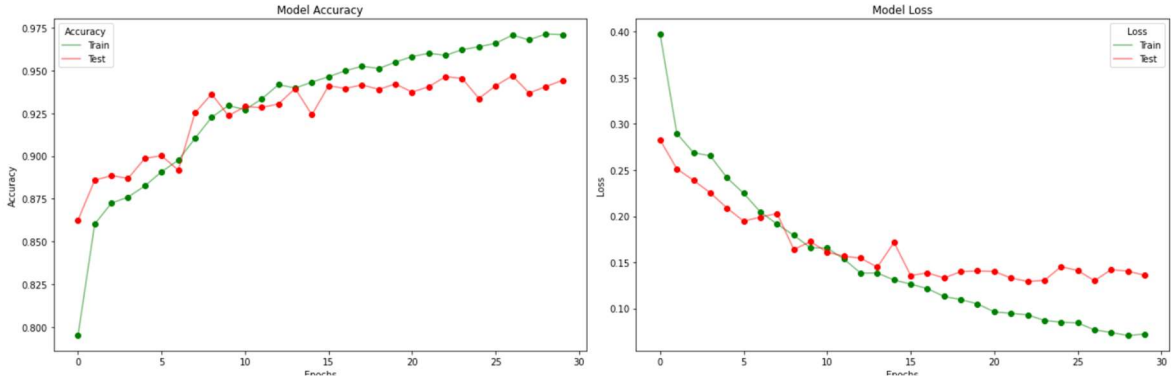


Figure 5.6: CNN-ENHAN Accuracy/loss graph

Graphs for Method 2 (ResNet50) is shown below:

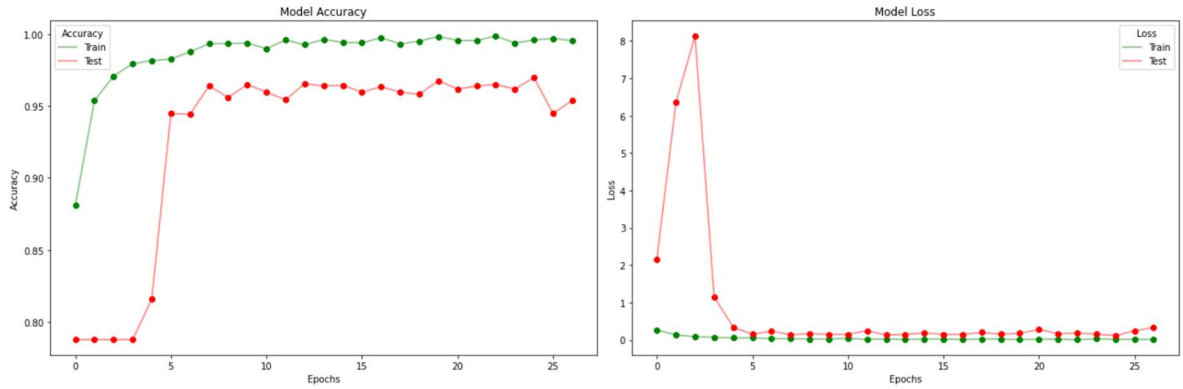


Figure 5.6: ResNet50 Accuracy/loss graph

6. Conclusion and recommendations for future

Recent years have seen an increase in popularity of photography due to the increased availability of cameras. Images play a critical role in our lives and have evolved into an indispensable method of transmitting information due to the ease with which the general public knows and understands them. There are numerous tools available for editing images; while these tools are primarily designed to enhance images, they are frequently used to fabricate images in order to spread falsehoods.

As a result, image forgery has turned into a massive problem. We present a novel image fraud detection system based on neural networks and deep learning, with a particular emphasis on the CNN architecture method. To obtain good results, the proposed method makes use of a CNN architecture that combines image compression changes (Error level analysis). To train the model, we use the difference between the original and recompressed images. The suggested technique is capable of efficiently detecting image forgeries involving image splicing and copy-move techniques. The findings of the studies are extremely positive, indicating that the overall validation accuracy is 94.43 percent when using a predetermined epoch of 30. As indicated by the confusion matrices and average accuracies, ResNet-50 outperformed MobileNetV2 in terms of accuracy and loss performance. The ResNet-50 achieves accuracy in 30 epochs, while the MobileNet achieves accuracy in epochs.

We plan to extend our technique for image forgery localization in the future. We will also combine the suggested technique with other known image localization techniques to improve their performance in terms of accuracy and reduce their time complexity. We will enhance the proposed technique to handle spoofing [50] as well. The present technique requires image

resolution to be a minimum of 128 × 128, so we will enhance the proposed technique to work well for tiny images.

We will also be developing a challenging extensive image forgery database to train deep learning networks for image forgery detection.

- According to the complexities of the datasets, the number of convolution and pooling layers could be increased.
- Could apply the approaches on different and variety of dataset or modified the current model with simple changes in the area of training algorithms and/or at pre-processing stage.
- To broaden the approaches could be applied on videos which are the collection of frames

References

Abhishek, Jindal, N., 2021. Copy move and splicing forgery detection using deep convolution neural network, and semantic segmentation. *Multimed Tools Appl* **80,** 3571–3599. Available at: https://link.springer.com/article/10.1007%2Fs11042-020-09816-3 #citeas

Ali, S.S.; Ganapathi, I.I.; Vu,N.-S.; Ali, S.D.; Saxena, N.; Werghi, N.Image Forgery Detection Using Deeplearning by Recompressing Images. Electronics 2022, 11, 403. Available at: https://doi.org/10.3390/electronics11030403 .

Ali, S.S.; Iyappan, G.I.; Prakash, S. Fingerprint Shell construction with impregnable features. J. Intell. Fuzzy Syst. 2019, 36, 4091–4104. [CrossRef]

Asghar, K., Sun, X., Rosin, P. L., Saddique, M., Hussain, M., and Habib, Z. 2019. Edge–texture feature-based image forgery detection with cross-dataset evaluation. *Machine Vision and Applications*, *30*(7), pp. 1243-1262. http://orca.cf.ac.uk/126350/1/Cross-Dataset%20Evaluation%20of%20Image%20Forgery%20Detection.pdf

Bammey, Q., Nikoukhah, T., Gardella, M., von Gioi, R.G., Colom, M. and Morel, J.M., 2022. Non-Semantic Evaluation of Image Forensics Tools: Methodology and Database. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 3751-3760). Available at: https://openaccess.thecvf.com/content/WACV2022/papers/Bammey_Non-Semantic_Evaluation_of_Image_Forensics_Tools_Methodology_and_Database_WACV_2022_paper.pdf.

A. Tuama, F. Comby, M. Chaumont. **Camera model identification with the use of deep convolutional neural networks.** Information Forensics and Security (WIFS), 2016 IEEE International Workshop on, IEEE (2016), pp. 1-6

Bammey, Q., von Gioi, R.G. and Morel, J.M., 2022. Forgery Detection by Internal Positional Learning of Demosaicing Traces. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 328-338). Available at: https://openaccess.thecvf.com/content/WACV2022/papers/Bammey_Forgery_Detection_by_Internal_Positional_Learning_of_Demosaicing_Traces_WACV_2022_paper.pdf.

Bao, Z. and Xue, R., 2022. Survey on deep learning applications in digital image security. *Optical Engineering*, *60*(12), p.120901. Available at: https://www.spiedigitallibrary.org/journals/optical-engineering/volume-60/issue-12/120901/Survey-on-deep-learning-applications-in-digital-image-security/10.1117/1.OE.60.12.120901.pdf.

B. Diallo, T. Urruty, P. Bourdon, C. Fernandez-Maloigne. **Improving Robustness of Image Tampering Detection for Compression** (2019), pp. 387-398

Blacksmith, S.S., Rasheed, A., Kvamsdal, T. and San, O., 2022. Deep neural network-enabled corrective source term approach to hybrid analysis and modeling. *Neural Networks*, *146*, pp.181-199. Available at: https://www.sciencedirect.com/science/article/pii/S0893608021004494.

Bouktif, S., Fiaz, A., Ouni, A. and Serhani, M.A., 2018. Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches. Energies, 11(7), p.1636.

Brownlee, Jason. Deep learning for computer vision: image classification, object detection, and face recognition in python. Machine Learning Mastery, 2019.

Cao, Gang, Yao Zhao, Rongrong Ni, and Alex C. Kot. "Unsharp masking sharpening detection via overshoot artifacts analysis." IEEE Signal Processing Letters 18, no. 10 (2011): 603-606.

Cao, Gang, Yao Zhao, and Rongrong Ni. "Detection of image sharpening based on histogram aberration and ringing artifacts." In 2009 IEEE International Conference on Multimedia and Expo, pp. 1026-1029.IEEE, 2009.

Chelghoum R., Ikhlef A., Hameurlaine A., Jacquir S. (2020) Transfer Learning Using Convolutional Neural Network Architectures for Brain Tumor Classification from MRI Images. In: Maglogiannis I., Iliadis L., Pimenidis E. (eds) Artificial Intelligence Applications and Innovations. AIAI 2020. IFIP Advances in Information and Communication Technology, vol 583. Springer, Cham. https://doi.org/10.1007/978-3-030-49161-1_17

Chen, Y., Tang, X., Qi, X., Li, C.G. and Xiao, R., 2022. Learning graph normalization for graph neural networks. *Neurocomputing*. Available at: https://arxiv.org/pdf/2009.11746.

Deep Kaur, C. and Kanwal, N., 2019. An analysis of image forgery detection techniques. Statistics, Optimization & Information Computing, 7(2), pp.486-500.

Corel Database, “http://corel.digitalriver.com/,”

Di, ZHOU, ZHUANG, X. and Hongfu, Z.U.O., 2022. A hybrid deep neural network based on multi-time window convolutional bidirectional LSTM for civil aircraft APU hazard identification. *Chinese Journal of Aeronautics*, *35*(4), pp.344-361. Available at: https://www.sciencedirect.com/science/article/pii/S1000936121001229.

Ding, Feng, Guopu Zhu, Weiqiang Dong, and Yun-Qing Shi. "An efficient weak sharpening detection method for image forensics." Journal of Visual Communication and Image Representation 50 (2018): 93-99.

Dombrowski, A.K., Anders, C.J., Müller, K.R. and Kessel, P., 2022. Towards robust explanations for deep neural networks. *Pattern Recognition*, *121*, p.108194. Available at: https://www.sciencedirect.com/science/article/pii/S0031320321003769.

Fahime Hakimi, Mahdi Hariri, Farhad GharehBaghi, “Image Splicing Forgery Detection using Local Binary Pattern and Discrete Wavelet Transform”, 2nd international conference on KBEI, IEEE, 2015.

F. Marra, G. Poggi, C. Sansone, L. Verdoliva. **Evaluation of residual-based local features for camera model identification.** International Conference on Image Analysis and Processing, Springer (2015), pp. 11-18

FotoForensics. (n.d.). FotoForensics. https://fotoforensics.com/tutorial-ela.php.

Ghoneim, A., Muhammad, G., Amin, S.U. and Gupta, B., 2018. Medical image forgery detection for smart healthcare. IEEE Communications Magazine, 56(4), pp.33-37.

Hawkins, C., Liu, X. and Zhang, Z., 2022. Towards compact neural networks via end-to-end training: A bayesian tensor approach with automatic rank determination. *SIAM Journal on Mathematics of Data Science*, *4*(1), pp.46-71. Available at: https://arxiv.org/pdf/2010.08689.pdf.

Hung, Y.C., Zhao, Y.X. and Hung, W.C., 2022. Development of an Underground Tunnels Detection Algorithm for Electrical Resistivity Tomography Based on Deep Learning. *Applied Sciences*, *12*(2), p.639. Available at: https://www.mdpi.com/2076-3417/12/2/639/pdf.

(PDF) Image Forgery Detection: Survey and Future Directions. https://www.researchgate.net/publication/332633501_Image_Forgery_Detection_Survey_and_Future_Directions

Hussain, Mahbub, Jordan J. Bird, and Diego R. Faria. "A study on cnn transfer learning for image classification." In UK Workshop on computational Intelligence, pp. 191-202. Springer, Cham, 2018.

Jalab, H.A., Alqarni, M.A., Ibrahim, R.W. and Almazroi, A.A., 2022. A Novel Pixel's Fractional Mean-Based Image Enhancement Algorithm for Better Image Splicing Detection. *Journal of King Saud University-Science*, p.101805. Available at: https://www.sciencedirect.com/science/article/pii/S1018364721004675.

J. Dong, W. Wang, and T. Tan, "CASIA image tampering detection evaluation database," in 2013 IEEE China Summit and International Conference on Signal and Information Processing, ChinaSIP 2013 - Proceedings, 2013, pp. 422–426. https://doi.org/10.1109/ChinaSIP.2013.6625374

Jin, P., Lai, T., Lai, R. and Dong, B., 2022. NPTC-net: Narrow-Band Parallel Transport Convolutional Neural Networks on Point Clouds. *Journal of Scientific Computing*, *90*(1), pp.1-20. Available at: https://www.sciencedirect.com/science/article/pii/S0893608021004494.

Jeronimo DC, Borges YCC, Coelho L dos S. Image forgery detection by semi-automatic wavelet soft-Thresholding with error level analysis. Expert Systems with Applications. 2017; 85: 348–56

Kadam, K.D., Ahirrao, S. and Kotecha, K., (2022). Efficient Approach towards Detection and Identification of Copy Move and Image Splicing Forgeries Using Mask R-CNN with MobileNet V1. *Computational Intelligence and Neuroscience*, *2022*. Available at: https://www.hindawi.com/journals/cin/2022/6845326/.

Kim, D.K. and Kim, K.S., 2022. Generalized Facial Manipulation Detection With Edge Region Feature Extraction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 2828-2838). Available at: https://openaccess.thecvf.com/content/WACV2022/papers/Kim_Generalized_Facial_Manipulation_Detection_With_Edge_Region_Feature_Extraction_WACV_2022_paper.pdf.

Krishna, S., Krishnamoorthy, S. and Bhavsar, A., 2022. Stain Normalized Breast Histopathology Image Recognition using Convolutional Neural Networks for Cancer Detection. *arXiv preprint arXiv:2201.00957*. Available at: https://arxiv.org/pdf/2112.13165.pdf.

Kumar, N. and Meenpal, T., 2022. Salient keypoint-based copy-move image forgery detection. *Australian Journal of Forensic Sciences*, pp.1-24. Available at: https://www.researchgate.net/publication/357838861_Salient_keypoint-based_copy-move_image_forgery_detection.

Kuruc, F., Binder, H. and Hess, M., 2022. Stratified neural networks in a time-to-event setting. *Briefings in Bioinformatics*, *23*(1), p.bbab392. Available at: https://www.biorxiv.org/content/biorxiv/early/2021/02/02/2021.02.01.429169.full.pdf.

Kuznetsov, A. (2019, November). Digital image forgery detection using deep learning approach. In *Journal of Physics: Conference Series* (Vol. 1368, No. 3, p. 032028). IOP Publishing.https://iopscience.iop.org/article/10.1088/1742-6596/1368/3/032028/pdf.

Kwon, M.J.; Yu, I.J.; Nam, S.H.; Lee, H.K. CAT-Net: Compression Artifact Tracing Network for Detection and Localization of Image Splicing. In Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 5–9 January 2021; pp. 375–384

Le-Tien, T., Phan-Xuan, H., Nguyen-Chinh, T. and Do-Tieu, T., 2019. Image forgery detection: A low computational-cost and effective data-driven model. International Journal of Machine Learning and Computing, 9(2).

Li, Y., and Zhou, J. 2018. Fast and effective image copy-move forgery detection via hierarchical feature point matching. *IEEE Transactions on Information Forensics and Security*, *14*(5), pp. 1307-1322.https://ieeexplore.ieee.org/abstract/document/8501936/

Lim, J. and Psaltis, D., 2022. MaxwellNet: Physics-driven deep neural network training based on Maxwell's equations. *APL Photonics*, *7*(1), p.011301. Available at: https://aip.scitation.org/doi/full/10.1063/5.0071616.

Liu, H. and Lang, B., 2019. Machine learning and deep learning methods for intrusion detection systems: A survey. applied sciences, 9(20), p.4396.

Ma, W., Tu, X., Luo, B. and Wang, G., 2022. Semantic clustering based deduction learning for image recognition and classification. *Pattern Recognition*, *124*, p.108440. Available at: https://arxiv.org/pdf/2112.13165.

Mandal, D., Medya, S., Uzzi, B. and Aggarwal, C., 2022. MetaLearning with Graph Neural Networks: Methods and Applications. *ACM SIGKDD Explorations Newsletter*, *23*(2), pp.13-22. Available at: https://dl.acm.org/doi/pdf/10.1145/3510374.3510379.

Mandru, D.B., Aruna Safali, M., Raghavendra Sai, N. and Sai Chaitanya Kumar, G., 2022. Assessing deep neural network and shallow for network intrusion detection systems in cyber security. In *Computer Networks and Inventive Communication Technologies* (pp. 703-713). Springer, Singapore. Available at: https://www.researchgate.net/profile/N-Raghavendra-Sai/publication/351691807_Assessing_Deep_Neural_Network_and_Shallow_for_Network_Intrusion_Detection_Systems_in_Cyber_Security/links/60a4fdf7a6fdcc3f301d804b/Assessing

-Deep-Neural-Network-and-Shallow-for-Network-Intrusion-Detection-Systems-in-Cyber-Security.pdf.

Marra, F., Gragnaniello, D., Verdoliva, L. and Poggi, G., 2020. A full-image full-resolution end-to-end-trainable CNN framework for image forgery detection. IEEE Access, 8, pp.133488-133502.

Marra, F., Gragnaniello, D., Verdoliva, L., and Poggi, G. 2020. A full-image full-resolution end-to-end-trainable CNN framework for image forgery detection. *IEEE Access*, *8*, 133488-133502.https://ieeexplore.ieee.org/iel7/6287639/8948470/09142188.pdf

Matern, F., Riess, C., and Stamminger, M. 2019. Gradient-based illumination description for image forgery detection. *IEEE Transactions on Information Forensics and Security*, *15*, pp. 1303-1317.https://ieeexplore.ieee.org/abstract/document/8812683/

Mayer, O., and Stamm, M. C. 2018. Accurate and efficient image forgery detection using lateral chromatic aberration. *IEEE transactions on information forensics and security*, *13*(7), pp. 1762-1777.https://misl.ece.drexel.edu/wp-content/uploads/2018/02/Mayer_TIFS_2018.pdf

Mazaheri, G. and Roy-Chowdhury, A.K., 2022. Detection and Localization of Facial Expression Manipulations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 1035-1045). Available at: https://openaccess.thecvf.com/content/WACV2022/papers/Mazaheri_Detection_and_Localization_of_Facial_Expression_Manipulations_WACV_2022_paper.pdf.

Meena, K. B., and Tyagi, V. 2019. A copy-move image forgery detection technique based on Gaussian-Hermite moments. *Multimedia Tools and Applications*, *78*(23), pp. 33505-33526.https://link.springer.com/article/10.1007/s11042-019-08082-2

Meena, K. B., and Tyagi, V. 2019. Image forgery detection: survey and future directions. In *Data, Engineering and applications* (pp. 163-194). Springer, Singapore.https://www.researchgate.net/profile/Vipin-Tyagi-2/publication/332633501_Image_Forgery_Detection_Survey_and_Future_Directions/links/5d89ac18a6fdcc8fd61b0a63/Image-Forgery-Detection-Survey-and-Future-Directions.pdf

Meena, K. B., and Tyagi, V. 2020. A copy-move image forgery detection technique based on tetrolet transform. *Journal of Information Security and Applications*, *52*, p. 102481.https://www.sciencedirect.com/science/article/pii/S221421261930660X

Meena, K. B., &Tyagi, V. (2020). A hybrid copy-move image forgery detection technique based on Fourier-Mellin and scale-invariant feature transforms. *Multimedia Tools and*

Applications, *79*(11), 8197-8212.https://link.springer.com/article/10.1007/s11042-019-08343-0

Mohammed, T.M., Bunk, J., Nataraj, L., Bappy, J.H., Flenner, A., Manjunath, B.S., Chandrasekaran, S., Roy-Chowdhury, A.K. and Peterson, L.A., 2018. Boosting image forgery detection using resampling features and copy-move analysis. Electronic Imaging, 2018(7), pp.118-1.

Nirmala, G. and Thyagharajan, K.K., 2019, April. A modern approach for image forgery detection using BRICH clustering based on normalised mean and standard deviation. In 2019 International Conference on Communication and Signal Processing (ICCSP) (pp. 0441-0444). IEEE.

Normalization (image processing) - Wikipedia. Available at: https://en.wikipedia.org/wiki/Normalization_%28image_processing%29

Ohn, I. and Kim, Y., 2022. Nonconvex sparse regularization for deep neural networks and its optimality. *Neural Computation*, *34*(2), pp.476-517. Available at: https://arxiv.org/pdf/2003.11769.

Parveen, A., Khan, Z. H., and Ahmad, S. N. 2019. Block-based copy-move image forgery detection using DCT. *Iran Journal of Computer Science*, *2*(2), pp. 89-99.https://link.springer.com/article/10.1007/s42044-019-00029-y

Popescu, D., El-Khatib, M., El-Khatib, H. and Ichim, L., 2022. New Trends in Melanoma Detection Using Neural Networks: A Systematic Review. *Sensors*, *22*(2), p.496. Available at: https://www.mdpi.com/1424-8220/22/2/496/pdf.

Prakash, C. S., Kumar, A., Maheshkar, S., and Maheshkar, V. 2018. An integrated method of copy-move and splicing for image forgery detection. *Multimedia Tools and Applications*, *77*(20), pp. 26939-26963.https://link.springer.com/article/10.1007/s11042-018-5899-3

Prakash, C.S., Kumar, A., Maheshkar, S. and Maheshkar, V., 2018. An integrated method of copy-move and splicing for image forgery detection. Multimedia Tools and Applications, 77(20), pp.26939-26963.

Roy, A.M., Bose, R. and Bhaduri, J., 2022. A fast accurate fine-grain object detection model based on YOLOv4 deep neural network. *Neural Computing and Applications*, pp.1-27. Available at: https://arxiv.org/pdf/2111.00298.pdf.

Sadeghi, S., Dadkhah, S., Jalab, H. A., Mazzola, G., and Uliyan, D. 2018. State of the art in passive digital image forgery detection: copy-move image forgery. *Pattern Analysis and*

Applications, *21*(2), pp. 291-306.https://www.researchgate.net/profile/Giuseppe-Mazzola-2/publication/322074799_State_of_the_art_in_passive_digital_image_forgery_detection_copy-move_image_forgery/links/5a9fe19e45851543e6352c1f/State-of-the-art-in-passive-digital-image-forgery-detection-copy-move-image-forgery.pdf

Sadeghi, S., Dadkhah, S., Jalab, H.A., Mazzola, G. and Uliyan, D., 2018. State of the art in passive digital image forgery detection: copy-move image forgery. Pattern Analysis and Applications, 21(2), pp.291-306.

Salahuddin, Z., Woodruff, H.C., Chatterjee, A. and Lambin, P., 2022. Transparency of deep neural networks for medical image analysis: A review of interpretability methods. *Computers in biology and medicine*, *140*, p.105111. Available at: https://www.sciencedirect.com/science/article/pii/S0010482521009057.

Salim, M.Z., Abboud, A.J. and Yildirim, R., 2022. A Visual Cryptography-Based Watermarking Approach for the Detection and Localization of Image Forgery. *Electronics*, *11*(1), p.136. Available at: https://www.mdpi.com/2079-9292/11/1/136/pdf.

Shaha, Manali, and Meenakshi Pawar. "Transfer learning for image classification." In 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 656-660. IEEE, 2018.

Shin, Hoo-Chang, Holger R. Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M. Summers. "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning." IEEE transactions on medical imaging 35, no. 5 (2016): 1285-1298.

Sudiatmika, I.B.K. and Rahman, F., 2019. Image forgery detection using error level analysis and deep learning. Telkomnika, 17(2), pp.653-659.

Suissa, O., Elmalech, A. and Zhitomirsky-Geffet, M., 2022. Text analysis using deep neural networks in digital humanities and information science. *Journal of the Association for Information Science and Technology*, *73*(2), pp.268-287. Available at: https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/asi.24544.

Sun, Y., Ni, R. and Zhao, Y., 2022. MFAN: Multi-Level Features Attention Network for Fake Certificate Image Detection. *Entropy*, *24*(1), p.118. Available at: https://www.mdpi.com/1099-4300/24/1/118/htm.

Tang, J., Wang, Y., Fu, S., Liu, B. and Liu, W., 2022. A graph convolutional neural network model with Fisher vector encoding and channel-wise spatial-temporal aggregation for skeleton-based action recognition. *IET Image Processing*. Available at: https://ietresearch.onlinelibrary.wiley.com/doi/pdfdirect/10.1049/ipr2.12422.

T. J. de Carvalho, C. Riess, E. Angelopoulou, H. Pedrini and A. de Rezende Rocha, Exposing Digital Image Forgeries by Illumination Color Classification," in IEEE Transactions on Information Forensics and Security, vol. 8, no. 7, pp. 1182-1194, July 2013, DOI: 10.1109/TIFS.2013.2265677.

T. S. Gunawan, S. A. M. Hanafiah, M. Kartiwi, N. Ismail, N. F. Za'bah, and A. N. Nordin, Development of photo forensics algorithm by detecting photoshop manipulation using error level analysis," Indones. J. Electr. Eng. Comput. Sci., vol. 7, no. 1, pp. 131–137, Jul. 2017. http://doi.org/10.11591/ijeecs.v7.i1.pp131-137

Tulbure, A.A., Tulbure, A.A. and Dulf, E.H., 2022. A review on modern defect detection models using DCNNs–Deep convolutional neural networks. *Journal of Advanced Research*, *35*, pp.33-48. Available at: https://www.sciencedirect.com/science/article/pii/S2090123221000643.

Walia, S., and Kumar, K. 2019. Digital image forgery detection: a systematic scrutiny. *Australian Journal of Forensic Sciences*, *51*(5), pp. 488-526.https://www.researchgate.net/profile/Savita-Walia/publication/323574573_Digital_image_forgery_detection_a_systematic_scrutiny/links/5a9f90400f7e9badd99e8cc1/Digital-image-forgery-detection-a-systematic-scrutiny.pdf

Wei, L., Luo, Y., Xu, L., Zhang, Q., Cai, Q., and Shen, M., 2022. Deep Convolutional Neural Network for Rice Density Prescription Map at Ripening Stage Using Unmanned Aerial Vehicle-Based Remotely Sensed Images. *Remote Sensing*, *14*(1), p.46. Available at: https://www.mdpi.com/2072-4292/14/1/46/pdf.

Wei, S., Chen, Y., Zhou, Z. and Song, G., 2022. A quantum convolutional neural network on NISQ devices. *AAPPS Bulletin*, *32*(1), pp.1-11. Available at: https://link.springer.com/article/10.1007/s43673-021-00030-3.

Wu, Y.; Abd Almageed, W.; Natarajan, P. ManTra-Net: Manipulation Tracing Network for Detection and Localization of Image Forgeries with Anomalous Features. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9535–9544.

Xiao, B.; Wei, Y.; Bi, X.; Li, W.; Ma, J. Image splicing forgery detection combining coarse to refined convolutional neural network and adaptive clustering. Inf. Sci. 2020, 511, 172–191. [CrossRef]

Yarlagadda, S. K., Güera, D., Bestagini, P., Maggie Zhu, F., Tubaro, S., and Delp, E. J. 2018. Satellite image forgery detection and localization using gan and one-class classifier. *Electronic Imaging*, *2018*(7), pp. 214-1.https://www.ingentaconnect.com/contentone/ist/ei/2018/00002018/00000007/art00012?crawler=true&mimetype=application/pdf

Yarlagadda, S.K., Güera, D., Bestagini, P., Maggie Zhu, F., Tubaro, S. and Delp, E.J., 2018. Satellite image forgery detection and localization using gan and one-class classifier. Electronic Imaging, 2018(7), pp.214-1.

Yarlagadda, S.K., Güera, D., Bestagini, P., Maggie Zhu, F., Tubaro, S. and Delp, E.J., 2018. Satellite image forgery detection and localization using gan and one-class classifier. Electronic Imaging, 2018(7), pp.214-1.

Zhang, Z., Chen, M., Backes, M., Shen, Y., and Zhang, Y., 2022. Inference attacks against graph neural networks. In *USENIX Security Symposium (USENIX Security). USENIX* (Vol. 13). Available at: https://www.usenix.org/system/files/sec22summer_zhang-zhikun.pdf.

Zhang, Z., Zhang, Y., Zhou, Z. and Luo, J., 2018, August. Boundary-based image forgery detection by fast shallow cnn. In 2018 24th International Conference on Pattern Recognition (ICPR) (pp. 2658-2663). IEEE.

Zhu, W., Xu, K., Darve, E., Biondi, B. and Beroza, G.C., 2022. Integrating deep neural networks with full-waveform inversion: Reparameterization, regularization, and uncertainty quantification. *Geophysics*, *87*(1), pp.R93-R109.
Available at: https://arxiv.org/pdf/2012.11149.pdf.

Z. J. Barad and M. M. Goswami, "Image Forgery Detection using Deep Learning: A Survey," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, pp. 571-576, DOI: 10.1109/ICACCS48705.2020.907440.

Appendices

Appendix 1: CNN accuracy percentage

```
Epoch 1/15
296/296 [==============================] - 53s 176ms/step - loss: 0.6658 - accuracy: 0.6040 - val_loss: 0.6823 - val_accuracy: 0.5701
Epoch 2/15
296/296 [==============================] - 55s 185ms/step - loss: 0.6381 - accuracy: 0.6312 - val_loss: 0.6639 - val_accuracy: 0.6218
Epoch 3/15
296/296 [==============================] - 56s 189ms/step - loss: 0.6194 - accuracy: 0.6440 - val_loss: 0.6649 - val_accuracy: 0.6250
Epoch 4/15
296/296 [==============================] - 60s 202ms/step - loss: 0.5935 - accuracy: 0.6758 - val_loss: 0.6537 - val_accuracy: 0.6133
Epoch 5/15
296/296 [==============================] - 67s 226ms/step - loss: 0.5709 - accuracy: 0.6852 - val_loss: 0.6529 - val_accuracy: 0.6129
Epoch 6/15
296/296 [==============================] - 66s 225ms/step - loss: 0.5491 - accuracy: 0.7045 - val_loss: 0.6715 - val_accuracy: 0.6139
Epoch 7/15
296/296 [==============================] - 69s 232ms/step - loss: 0.5246 - accuracy: 0.7266 - val_loss: 0.6545 - val_accuracy: 0.6022
Epoch 8/15
296/296 [==============================] - 67s 227ms/step - loss: 0.5012 - accuracy: 0.7429 - val_loss: 0.6556 - val_accuracy: 0.6215
```

Appendix 2: CNN with ELA accuracy percentage.

```
Epoch 15/30
236/236 [==============================] - 6s 27ms/step - loss: 0.2507 - accuracy: 0.8978 - val_loss: 0.2479 - val_accuracy: 0.8976
Epoch 16/30
236/236 [==============================] - 6s 27ms/step - loss: 0.2444 - accuracy: 0.8965 - val_loss: 0.2253 - val_accuracy: 0.8960
Epoch 17/30
236/236 [==============================] - 6s 27ms/step - loss: 0.2367 - accuracy: 0.9005 - val_loss: 0.2183 - val_accuracy: 0.9071
Epoch 18/30
236/236 [==============================] - 6s 27ms/step - loss: 0.2289 - accuracy: 0.9036 - val_loss: 0.2075 - val_accuracy: 0.9119
Epoch 19/30
236/236 [==============================] - 6s 27ms/step - loss: 0.2230 - accuracy: 0.9083 - val_loss: 0.2085 - val_accuracy: 0.9029
Epoch 20/30
236/236 [==============================] - 6s 27ms/step - loss: 0.2191 - accuracy: 0.9093 - val_loss: 0.2019 - val_accuracy: 0.9108
Epoch 21/30
236/236 [==============================] - 6s 27ms/step - loss: 0.2107 - accuracy: 0.9144 - val_loss: 0.2019 - val_accuracy: 0.9124
Epoch 22/30
236/236 [==============================] - 6s 27ms/step - loss: 0.2025 - accuracy: 0.9197 - val_loss: 0.2077 - val_accuracy: 0.9135
Epoch 23/30
236/236 [==============================] - 6s 27ms/step - loss: 0.1998 - accuracy: 0.9188 - val_loss: 0.1885 - val_accuracy: 0.9193
Epoch 24/30
236/236 [==============================] - 6s 27ms/step - loss: 0.1975 - accuracy: 0.9209 - val_loss: 0.1933 - val_accuracy: 0.9199
Epoch 25/30
236/236 [==============================] - 6s 27ms/step - loss: 0.1865 - accuracy: 0.9265 - val_loss: 0.1909 - val_accuracy: 0.9214
Epoch 26/30
236/236 [==============================] - 6s 28ms/step - loss: 0.1840 - accuracy: 0.9261 - val_loss: 0.1917 - val_accuracy: 0.9145
Epoch 27/30
236/236 [==============================] - 6s 27ms/step - loss: 0.1757 - accuracy: 0.9327 - val_loss: 0.2026 - val_accuracy: 0.9124
Epoch 28/30
236/236 [==============================] - 6s 27ms/step - loss: 0.1748 - accuracy: 0.9295 - val_loss: 0.1889 - val_accuracy: 0.9193
Epoch 29/30
236/236 [==============================] - 6s 27ms/step - loss: 0.1629 - accuracy: 0.9347 - val_loss: 0.1658 - val_accuracy: 0.9268
Epoch 30/30
236/236 [==============================] - 6s 27ms/step - loss: 0.1642 - accuracy: 0.9319 - val_loss: 0.1724 - val_accuracy: 0.9273
```

Appendix 3: CNN with ELA Sharpen accuracy percentage.

```
Epoch 16/30
236/236 [==============================] - 6s 26ms/step - loss: 0.1264 - accuracy: 0.9464 - val_loss: 0.1357 - val_accuracy: 0.9411
Epoch 17/30
236/236 [==============================] - 6s 25ms/step - loss: 0.1216 - accuracy: 0.9498 - val_loss: 0.1387 - val_accuracy: 0.9395
Epoch 18/30
236/236 [==============================] - 6s 25ms/step - loss: 0.1130 - accuracy: 0.9525 - val_loss: 0.1331 - val_accuracy: 0.9416
Epoch 19/30
236/236 [==============================] - 6s 25ms/step - loss: 0.1097 - accuracy: 0.9512 - val_loss: 0.1400 - val_accuracy: 0.9390
Epoch 20/30
236/236 [==============================] - 6s 25ms/step - loss: 0.1051 - accuracy: 0.9549 - val_loss: 0.1407 - val_accuracy: 0.9421
Epoch 21/30
236/236 [==============================] - 6s 26ms/step - loss: 0.0963 - accuracy: 0.9582 - val_loss: 0.1402 - val_accuracy: 0.9374
Epoch 22/30
236/236 [==============================] - 6s 25ms/step - loss: 0.0948 - accuracy: 0.9600 - val_loss: 0.1331 - val_accuracy: 0.9406
Epoch 23/30
236/236 [==============================] - 6s 25ms/step - loss: 0.0931 - accuracy: 0.9589 - val_loss: 0.1293 - val_accuracy: 0.9464
Epoch 24/30
236/236 [==============================] - 6s 26ms/step - loss: 0.0871 - accuracy: 0.9622 - val_loss: 0.1305 - val_accuracy: 0.9453
Epoch 25/30
236/236 [==============================] - 6s 26ms/step - loss: 0.0852 - accuracy: 0.9639 - val_loss: 0.1453 - val_accuracy: 0.9337
Epoch 26/30
236/236 [==============================] - 6s 25ms/step - loss: 0.0844 - accuracy: 0.9659 - val_loss: 0.1412 - val_accuracy: 0.9411
Epoch 27/30
236/236 [==============================] - 6s 26ms/step - loss: 0.0769 - accuracy: 0.9707 - val_loss: 0.1301 - val_accuracy: 0.9469
Epoch 28/30
236/236 [==============================] - 6s 25ms/step - loss: 0.0741 - accuracy: 0.9679 - val_loss: 0.1421 - val_accuracy: 0.9368
Epoch 29/30
236/236 [==============================] - 6s 26ms/step - loss: 0.0707 - accuracy: 0.9713 - val_loss: 0.1404 - val_accuracy: 0.9406
Epoch 30/30
236/236 [==============================] - 6s 25ms/step - loss: 0.0723 - accuracy: 0.9709 - val_loss: 0.1361 - val_accuracy: 0.9443
```

Appendix 4: MobileNetV2 accuracy percentage

```
Epoch 1/15
236/236 [==============================] - 24s 79ms/step - loss: 0.5237 - acc: 0.8237 - val_loss: 2.9531 - val_acc: 0.2845
Epoch 2/15
236/236 [==============================] - 18s 75ms/step - loss: 0.2289 - acc: 0.9360 - val_loss: 0.6977 - val_acc: 0.7028
Epoch 3/15
236/236 [==============================] - 18s 77ms/step - loss: 0.1362 - acc: 0.9602 - val_loss: 0.6158 - val_acc: 0.7866
Epoch 4/15
236/236 [==============================] - 18s 76ms/step - loss: 0.0974 - acc: 0.9729 - val_loss: 0.7066 - val_acc: 0.7877
Epoch 5/15
236/236 [==============================] - 18s 75ms/step - loss: 0.0671 - acc: 0.9796 - val_loss: 0.7730 - val_acc: 0.5844
Epoch 6/15
236/236 [==============================] - 18s 75ms/step - loss: 0.0516 - acc: 0.9849 - val_loss: 0.5222 - val_acc: 0.7500
Epoch 7/15
236/236 [==============================] - 18s 76ms/step - loss: 0.0579 - acc: 0.9818 - val_loss: 0.5688 - val_acc: 0.6943
Epoch 8/15
236/236 [==============================] - 18s 75ms/step - loss: 0.0573 - acc: 0.9842 - val_loss: 0.5001 - val_acc: 0.7723
Epoch 9/15
236/236 [==============================] - 18s 75ms/step - loss: 0.0362 - acc: 0.9882 - val_loss: 0.5844 - val_acc: 0.7776
Epoch 10/15
236/236 [==============================] - 18s 76ms/step - loss: 0.0357 - acc: 0.9902 - val_loss: 0.6319 - val_acc: 0.7187
Epoch 11/15
236/236 [==============================] - 20s 84ms/step - loss: 0.0464 - acc: 0.9874 - val_loss: 1.0404 - val_acc: 0.6900
Epoch 12/15
236/236 [==============================] - 20s 83ms/step - loss: 0.0494 - acc: 0.9881 - val_loss: 0.8661 - val_acc: 0.7861
Epoch 13/15
236/236 [==============================] - 18s 75ms/step - loss: 0.0417 - acc: 0.9894 - val_loss: 0.7273 - val_acc: 0.7282
Epoch 14/15
236/236 [==============================] - 18s 76ms/step - loss: 0.0259 - acc: 0.9936 - val_loss: 0.6144 - val_acc: 0.8142
Epoch 15/15
236/236 [==============================] - 18s 75ms/step - loss: 0.0250 - acc: 0.9934 - val_loss: 0.7491 - val_acc: 0.8025
```

Appendix 5: ResetNet50 accuracy percentage

```
Epoch 1/11
236/236 [==============================] - 55s 163ms/step - loss: 0.3024 - accuracy: 0.8711 - val_loss: 51.9041 - val_accuracy: 0.7877
Epoch 2/11
236/236 [==============================] - 37s 157ms/step - loss: 0.2271 - accuracy: 0.9145 - val_loss: 0.9362 - val_accuracy: 0.7877
Epoch 3/11
236/236 [==============================] - 36s 154ms/step - loss: 0.1731 - accuracy: 0.9316 - val_loss: 2.0884 - val_accuracy: 0.7877
Epoch 4/11
236/236 [==============================] - 36s 153ms/step - loss: 0.1732 - accuracy: 0.9281 - val_loss: 0.5053 - val_accuracy: 0.7877
Epoch 5/11
236/236 [==============================] - 36s 153ms/step - loss: 0.1501 - accuracy: 0.9367 - val_loss: 0.4507 - val_accuracy: 0.8063
Epoch 6/11
236/236 [==============================] - 36s 154ms/step - loss: 0.1377 - accuracy: 0.9469 - val_loss: 0.1817 - val_accuracy: 0.9140
Epoch 7/11
236/236 [==============================] - 40s 170ms/step - loss: 0.1133 - accuracy: 0.9545 - val_loss: 0.1928 - val_accuracy: 0.9172
Epoch 8/11
236/236 [==============================] - 46s 196ms/step - loss: 0.1035 - accuracy: 0.9561 - val_loss: 0.1876 - val_accuracy: 0.9480
Epoch 9/11
236/236 [==============================] - 42s 179ms/step - loss: 0.0924 - accuracy: 0.9620 - val_loss: 0.1078 - val_accuracy: 0.9538
Epoch 10/11
236/236 [==============================] - 51s 216ms/step - loss: 0.0910 - accuracy: 0.9628 - val_loss: 0.1676 - val_accuracy: 0.9475
Epoch 11/11
236/236 [==============================] - 37s 157ms/step - loss: 0.0828 - accuracy: 0.9673 - val_loss: 0.1073 - val_accuracy: 0.9554
```

Appendix 6: Python coding for image forgery detection

```python
def image_processing(path,quality):
  '''
  Function process the image in order to optimize them to tra
  '''

  # Opening Image form directory using path
  image = Image.open(path).convert('RGB')
  # saving orignal image by resizing it (Reducing Quality)
  image.save('temp_file.jpg', 'JPEG', quality = quality)

  # Loading without RGB and low quality
  temp_image = Image.open('temp_file.jpg')

  # Comparing differences
  ela_img = ImageChops.difference(image, temp_image)
  # GEtting vectors as text
  extrema = ela_img.getextrema()
  # Scaling
  max_diff = max([ex[1] for ex in extrema])
  if max_diff == 0:
      max_diff = 1
  scale = 255.0 / max_diff

  # Increasing brightness of the image
  ela_img = ImageEnhance.Brightness(ela_img).enhance(scale)

  return ela_img
```

Appendix 7: Coding for Normalization

Normalization

```
[28] X = asarray(X,dtype='float32')
     print(X.shape)
```

```
(9418, 49152)
```

```
[29] Y = asarray(Y,dtype='float32')
     print(Y.shape)
```

```
(9418,)
```

Reshape

```
[30] Y = tf.keras.utils.to_categorical(Y, 2)
     X = X.reshape(-1, 128, 128, 3)
```

```
[ ] print(X.shape)
```

```
(9418, 128, 128, 3)
```