UNIVERSITY OF WITWATERSRAND

Ahmed Mohamoud Jama

Supervisor: Prof. Michael Sears

MSC RESEARCH REPORT

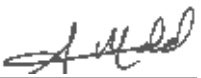
FAKE IMAGE DETECTION USING MACHINE LEARNING

2018

Declaration

I, Ahmed Mohamoud Jama, am a student registered for the degree of MSc. Computer Science in the academic year 2018. I hereby declare the following:

- I am aware that plagiarism (the use of someone else's work without their permission and/or without acknowledging the original source) is wrong.
- I confirm that the work submitted for assessment for the above degree is my own unaided work except where I have explicitly indicated otherwise.
- I have followed the required conventions in referencing the thoughts and ideas of others.

Signature:  **Date:** 23/03/2018

Acknowledgements

It my great pleasure to thank the all mighty God for giving me a healthy mind to complete this MSc Research Report. My special thanks go to my supervisor, Prof Michael Sears for his outstanding support and patience during the entire phase of this project. He always made himself available to answer any question I had, and having his support greatly influenced my research to be completed in time.

Abstract

The rapid growth of digital image processing technologies and editing software has given rise to large amounts of tampered images circulating in our daily lives. This undermines credibility and trustworthiness of digital images and also creates false beliefs in many real world situations. Hence it is generating a great demand for automatic forgery detection algorithms in order to determine the authenticity of a candidate image in a timely fashion. Many techniques and tools have been implemented to detect such type of forgery with the real image but because of increasing editing software every day, the problem is not solved yet. In this thesis, we will first introduce some information about digital photographs, their formats and how they are compressed, we will also introduce some important tools that were implemented to detect image forgery and how well these tools can detect faked or tampered images.

In this work we wish to test whether a convolutional neural network can be trained to efficiently detect image forgeries. The CNN will be trained first using two classes (real and fake), and then using the same CNN architecture with three classes (real, copy-paste and spliced images). The network will be trained and tested using the standard Casia 2.0 dataset. Then we will compare the convolutional neural network that we trained with some of the implemented tools that classifies images into real and fake.

We observe a very good performance after the training. That means the trained CNN network is able to recognize the tampered images at a maximum success rate of 85.63%. So the use of this application can be used as a false proof technique in digital authentication. It will also greatly reduce spreading of fake images through websites and through social media.

Contents

List of Figures

Chapter 1

Introduction

It is now easier to make high quality pictures and movies than ever before by using digital cameras and video software with high resolution at low cost. This leads to the extensive use of digital images for different purposes. Many services distribute pictures such as Flicker, MySpace and Google video [11]. Digital images are widely used as historical records and as evidence of real happenings in applications from police investigation, reports from journalists, insurance, medical and dental examinations, military and museum records to consumer photographs. One could say that today videos and digital images are more influential than words.

While digital images are widely used, their credibility has been challenged due to many fraudulent cases involving image forgeries. The fast development of digital image processing technologies and editing software such as GIMP, Photoshop, etc., make faking images much easier than before; they manipulate digital images to misrepresent the content of the original image and to deceive the viewers without leaving any obvious traces of such operations. And because of that it has become impossible to distinguish between a real digital camera image and a manipulated version of it. As a result of these problems, the credibility and trustworthiness of digital images is undermined in all aspects. It also creates false beliefs in many real world situations.

The old adage "don't believe everything you hear" is becoming "don't believe everything you see" [12]. A huge number of people have become victims of image forgery, specially celebrities and politicians whose pictures are photoshopped or tampered into fictional situations [13]. Traditionally image forensics is done simply by human inspection. Such approaches can succeed to get perfect detection and high quality analysis, but they typically require a significant amount of time and comprehensive human labour [5]. Nowadays, the number of tampered images circulated each day has far exceeded the amount that human inspection can handle, and editing software can splice different images together, copy and paste parts of the same image or graphically enhance images in a way that's difficult to detect by human observation. This encourages the need for automated detection tools to detect such types of image forgery.

The literature review in chapter 2 introduces some information about digital photographs, their formats and how they are compressed, that we need for our research. It also introduces some techniques used to forge digital images and also some techniques and methods related to image forgery detection. The research methodology in chapter 3 will outline the research problem and some important tools that were implemented to detect image forgery, and using these tools we want to check how well these tools can detect faked or tampered images.

The purpose of this work is to design and build a convolutional neural network (CNN) with the aim of detecting forgeries in images. We will train and test it on the Casia 2.0 image dataset. The CNN first classifies images into real and fake using two class classification methods. Secondly the same CNN architecture classifies images into real copy-paste and spliced images using three class classification, so we are using essentially the same network configuration but trained in two different ways. And at both stages we will print the accuracy of the network after training and testing. Finally we will compare this CNN to one of the standard implementations - Forensically - and check how well both are classifying images into real and fake.

Chapter 2

Literature Review

This chapter introduces some information about digital photographs and their creation pipeline, image formats, and how files are compressed. It also introduces some techniques used to forge digital images and their types, and also some techniques and methods related to image forgery detection.

2.1 Digital Photography

Digital photography is the process of capturing visible light from the electromagnetic spectrum, and saving it to a digital medium. A digital camera stores and records photographic images in digital form by using a sensor.

In digital photographs, the camera captures millions of 'dots' called pixels that a computer uses to display the image by dividing the screen into a corresponding grid of pixels each containing RGB (Red, Green, Blue) dots called sub-pixels. It then uses the values stored in the digital photograph to specify the brightness of each of the three sub-pixels and the combined brightnesses of the pixel are perceived as a single colour [14].

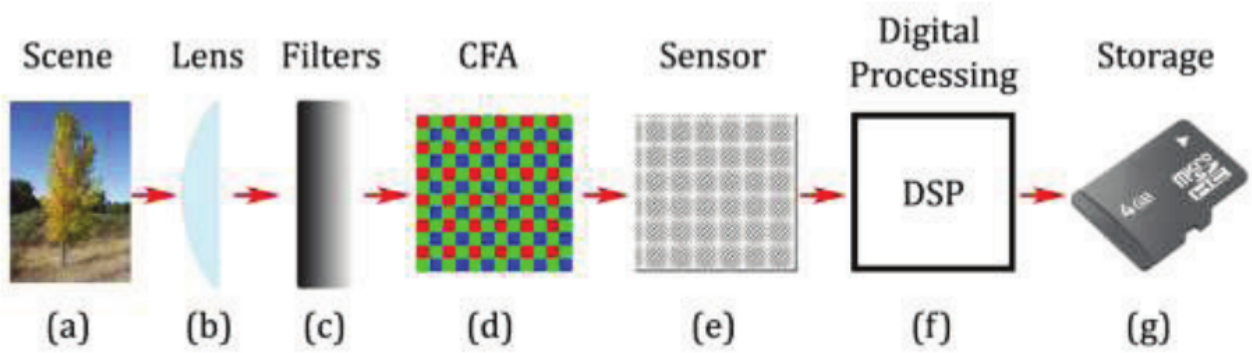


Figure 2.1: The digital image creation pipeline [1]

Figure 2.1 Illustrating the digital photography model. Light from the scene [figure 2.1(a)] is focused into the camera by an optical lens [figure 2.1(b)]. The lens is used to focus the image, and also to control the amount of light that enters the camera and how

much of the scene is photographed. The lenses can be divided into three categories: Wide-angle lenses that are used to capture a wide arc of the scene content and have shorter focal length than the focal length of a normal lens. Macro lenses that are used to capture the content that is very close to the lens. Telephoto lenses that capture images from far distance [15].

After the lens, the light passes through a series of filters that prepare it for conversion into the digital domain [figure 2.1(c)]. One of these filters is the anti-aliasing filter that sits on top of a camera's sensor and prevents high spatial frequencies, which correspond to very fine details in the scene, from passing through. However the anti-aliasing filter limits the frequencies that pass through to the sensor, and is often referred to as 'low-pass' filter, as it only allows lower frequencies to pass through to the sensor [1].

The image sensor consists of an array of picture elements, called pixels, that register the amount of light that falls onto them. The elements then convert the amount of light they received into the corresponding number of electrons, then the electrons are converted into voltage and then transformed into digital numbers. There are two types of sensors commonly used in cameras: CCD (Charge-Coupled Devise) and CMOS (Complementary Metal-oxide Semiconductor). Although the fundamental operation of each is similar, most digital cameras use a CMOS sensor, because it offers faster speed with lower power consumption [16].

Since pixels are not colour sensitive elements, a Colour Filter Array (CFA) is used to help distinguish between the different frequency ranges of light [figure 2.1(d)]. The colour filter array(CFA) is a mosaic of tiny colour filters placed on top of the monochrome image sensors, usually on top of CCD and CMOS sensor to separate the light by colour frequency onto each discrete pixel. There are many types of filter arrays, and one that is very common is the Bayer RGB colour filter array, that consists of a mosaic of red, green and blue. Since human eyes are more sensitive to green colours, the filter pattern is 50% green, 25% blue and 25% red [17].

After the CFA, the digital sensor captures the information by converting light intensity levels from a continuous signal to a finite number of intensity values through a process known as quantization. After the light intensity is converted by the sensor for each colour, a 'mosaicing' algorithm is needed to compute colour values for each of the three primary colours represented at each pixel location using a process known as interpolation [figure 2.1(f)], [18].

Next, a number of operations are performed by the internal camera processor, which can include white balancing and gamma correction [figure 2.1(f)]. The camera processes

the sensor information and creates a digital image file for storage onto a digital memory device [figure 2.1(g)].

2.2 File Compression

Image files are large compared to the other types of computer files. To reduce image size and to make them smaller and faster to load, most cameras use what we call image compression. Image compression shrinks the file size in bytes without degrading the image quality and also reduces the storage and transmission cost [19], and it can be classified into two types; lossless compression and lossy compression. The reduction in image file size has the below advantages:

1. Allows an images to take up less space in disk or memory.
2. Reduces the transmission time required for images to be sent over the Internet, and also transmission cost.
3. Reduces the time required to download images from Web pages/websites.

2.3 Image Formats

One of the most important work flow related decisions you make when capturing images is which image file format to use. An image file format is a standardized specification to encode information about an image into bits of data for storage. In a nutshell, an image saved and encoded in a known image format identifies itself as an image, and provides useful information such as its matrix size and bit depth [20]. There are plenty of image file formats, so it can be hard to know which file format type best suits your image needs. Some of them are good in printing like TIFF, while other are best for web graphics like JPEG, PNG. Here we will mention some common image file formats.

2.3.1 JPEG:

JPEG is the most popular file format used by almost every digital camera and other photographic image capture devices to store photos, and is widely used on the internet. JPEG stands for Joint Photographic Experts Group (the committee that created the file type) and is pronounced "jay-peg" [14]. JPEG is a lossy compressed file format, meaning that each time such file is created, a substantial amount of information is discarded to reduce storage required. All computer systems are able to read JPEG format, which is one reason why it is so popular [14].

2.3.2 RAW:

RAW is free of compression and contains all of the image data captured by the sensor without it being processed, that's why named Raw, because it is not yet processed or not ready to use as an image. Since it does not process the image, it leaves the editing process to the user. One thing to keep in mind is that RAW images are not always noticeably better. Where they shine is when you have exposure or white balance problems. Because RAW images have dramatically more information to work with, you can open up shadow areas, recover lost details in highlights, and make fine adjustments to colours [14].

2.3.3 TIFF/PSD:

These formats are lossless and most data for editing and utilizing purposes are preserved in these files which makes them large files. PSD which stands for (Photoshop Document), is a native photoshop format that can only be read by adobe applications like photoshop, It saves files in photoshop. TIFF (Tag Image File Format) is a common image format that is recognized by the vast majority of systems and applications, and can be recognized as a file with a ".tiff" or ".tif" file name. Tiff files are larger than the other files like RAW and JPEG. It can be saved using either 8 bits or 16 bits per colour. The difference between 8 and 16 bit is in image manipulations as 16 bit mode treats colours more accurately than 8 bit.

2.3.4 BMP:

The Windows Bitmap or BMP is an image file format used to store bitmap digital images, especially on Microsoft Windows OS. Typically, BMP files are large and uncompressed but the images are in high quality and rich in colour, it is also a simple structure and has wide acceptance in Windows programs.

2.4 Image Tampering

The use of digital photography has increased over the past few years, a trend which opens the door to new and creative ways to forge or tamper with digital images. Image tampering is defined as "adding or removing important features from an image without leaving any obvious traces of tampering" and thus image tampering is considered as intentional manipulation of images for malicious purposes [21]. The rapid growth of commercial and sometimes low cost or free of cost image editing softwares such as Photoshop, Photoscape, GIMP, Corel Paint Shop and PhotPlus dramatically increases the amount of tampered digital photographs circulated every day, without leaving any obvious visual traces of tampering [22]. They appear on television, in online forums, advertisements, movies, etc. Since news sources such as magazines and newspapers are supposed to report facts, enhanced or modified images used by the media deliberately or accidentally, could easily misrepresent real situations or mislead readers who believe them to be true. For example,

figure 2.2(a) shows a tampered photograph of two celebrities, movie stars Brad Pitt and Cher, that falsely implies their simultaneous presence at the same location [2].

Figure 2.2(b) is another tampered photograph which was widely circulated during the U.S. presidential election campaign in 2004. The photograph shows Jane Fonda and the U.S candidate John Kerry as if both were speakers at the same anti-Vietnam war protest in 1970, supposedly showing the strong anti-Vietnam war attitude of the then candidate John Kerry. This is when Senator Kerry was trying to get the democratic nomination. It later turned out to be a politically motivated fabricated composite of two different photographs taken at different times and places. The photograph of Jane Fonda was taken in 1972 speaking at Miami for a political march, while that of John Kerry was taken in Mineola, New York in June 1971 [3].

Figure 2.2: Examples of tampered images. (a) tampered photograph of two celebrities by movie stars Brad Pitt and Cher [2]; (b) a tampered photograph shows Jane Fonda and the U.S. candidate John Kerry in which both were speakers at an anti-Vietnam war protest in 1970 [3]; (c) shows a British soldier pointing his machine gun warning a group of Iraqi people urging them to seek cover from nearby fire [4]; (d) shows O. J. Simpson's photograph with skin colour darkened [5].

Figure 2.2(c) shows a British soldier pointing his machine gun and warning a group of Iraqi people to seek cover from nearby fire. It was first published on the front page of the Los Angeles Times in 2003, showing a brutal public image of the British Army [4].

Figure 2.2(d) shows a photograph of O. J. Simpson on the covers of both TIME magazine and Newsweek magazine in June 27, 1994 with Simpson's skin colour deliberately darkened [5].

INTERNET

Spies turn to high-tech info ops

PCs, Internet used for manipulating images, public opinion

ALTERED IMAGE

BY DANIEL VERTON

Federal intelligence agencies are studying ways to use computers and the Internet, rather than just leaflets and radio broadcasts, to shape and disseminate information designed to sway public opinion in the world's hot spots.

As part of its so-called "perception management" program, the intelligence community has for decades created misinformation to trigger political change without direct political or military involvement in countries where the United States has vested interests, such as Iraq and North Korea.

Acting on congressional recommendations to bolster research and development in information technology, intelligence agencies are turning to PCs to develop more sophisticated means of manipulating and delivering digital photos, video clips and recorded sound to portray fictitious events in hopes of provoking desirable outcomes.

1 2 3

Altered Egos

The top photo, of a fictitious meeting between President Clinton and Saddam Hussein, was generated by digitally altering and combining elements from three photos, shown below the altered image. Intelligence agencies plan to use similar techniques to create images and then disseminate them via the Internet in their efforts to influence events in foreign countries, such as Iraq.

(a) A Spliced image of Bill Clinton with Saddam Hussein[23].

(b) Photograph released by the news websites and public relations arm of Iran's Revolutionary Guards [24]

Figure 2.3: Tampered Figures

Figure 2.3(a) Upper part shows a photo of a fictitious meeting between president Saddam Hussein and president Bill Clinton. This photograph was generated by digitally altering and combining parts from three photos that we will see later when we are talking about analyzing methods of the tampered images. Intelligence agencies plan to use similar techniques to create images and then disseminate them via the internet in their efforts to influence events in foreign countries, such as Iraq [23].

Figure 2.3(b) Upper part displays a photograph released by the Iran news websites and also used by various western media including L.A Times, Chicago Tribune and New York Times. During the test firing of missiles, four missiles were supposed to be tested, but one missile on the ground has failed, so in order to cover up the failed one, the photograph has been tampered to show four missiles instead of three [24].

In the fields of criminal investigation, forensics, medical imaging, E-Commerce, surveillance systems, journalism and industrial photography, authenticity of digital images is of utmost importance. In order to detect image forgeries, it is important to understand the techniques and methods used to manipulate images.

2.4.1 Types Of Digital Image Tampering

There are many digital image tampering techniques and the most commonly used techniques are:

2.4.1.1 Copy-Paste Cloning:

Copy-Paste or copy-move attack is a common type of image tampering technique used to manipulate images. This technique copies part of the original image and pastes it onto some other part of the same image to hide some objects in the scene or to create more instances of some specific objects in an image. This technique uses some operations such as rotation, scaling and sometimes adding some noise to blend the pasted region with the original area, or to reduce the irregularities between the two regions. If carefully done, this kind of manipulation is difficult to detect by the human eye [25].

Figure 2.4: Images on the left are clones of the right images [6].

In figure 2.4 image (a) is a clone of image (b). The picture of the man in image (b) is concealed in image (a) by copy-pasting and blending a part of the scenery. Also, in figure 2.4 image (c) is a clone of image (d) where another gate is recreated by copy-pasting a part of the original image [6].

2.4.1.2 Image Splicing:

Image splicing is also one of the most common types of image tampering techniques. It copies a particular region of one image and pastes it into another image to create a new fake image, so the new image is a composite of one or more different images that are

combined to form a new image. To disguise the new copied region with the surrounding area, image splicing uses blending techniques such as blurring the edges, utilizing cloning, and reducing the contrast [26].

Figure 2.3 (a) Shows a forged image using image splicing technique. The image is a composite of three different images that you can see at the bottom which are numbered 1, 2 and 3. The White House image(in number 2), is re-scaled and blurred to create a background on which images of Bill Clinton and Saddam Hussein(numbered 1 and 3) are pasted [26].

2.4.1.3 Image Retouching:

The third technique of image tampering is called "Image Retouching", that is less harmful than the other image forgery techniques (copy-paste and splicing). Image retouching can either enhance or reduce features of the original image to make the image more attractive, and without changing its real meaning. This technique of tampering is used mostly by the magazine editors to make their images more attractive [21].

2.4.1.4 Re-size:

This operation is used to enlarge or shrink image size or sometimes part of an image. Re-sizing changes the size at which the image will print, but does not change the number of pixel in the image.

2.4.1.5 Cropping:

It is a technique to remove the outer parts (cut away some at the edges) of an image or to remove non-important aspects of an image.

2.5 Techniques to Analyze Images

2.5.1 Human Observation

To detect tampered images the simplest way is to use human observation or physical rules to detect changed regions in the image. To identify that, we can use items within the image that are:

- Shadow and Lighting. Shadow and sharp highlights shows light source and light direction, so when more images are concatenated into one image, they can have inconsistent shadows or may not have the same lighting. As we can see in figure 2.2(a) the shadow of Brad Pitt is at his right shoulder while the shadow of Cher is at her left shoulder.

- Scale and Floating. Merging too many images may cause inconsistent sizes. Also splicing a person into an image may show that person is not rooted on the ground or is floating.

- Items in the image. Sometimes items in the image may tell you more about when and where the image was taken. For example clocks and calendars can tell you date and time zone or disclose time. Other material in the image like electrical outlets and currency are different by the continent so using that you can identify the location [11].

2.5.2 Image Enhancements

Human observations can recognize items in the image, but many items need image enhancement methods to clarify elements within an image (or regions within the image). The common enhancement methods are:

- **Brightness and Contrast.** These algorithms make the image darker or brighter, or can separate brightest and darkest areas of the image. So to show a hidden object they can make dark area lighter, or tone down the brightness from a washed-out picture.

- **Sharpen and Blur.** Sharpening enhances the definition of edges in an image, so items that are blurred or not focused due to motion can be corrected using sharpening.

- **Histograms and Normalization.** Advanced tools allow the better viewing of colour ranges. So if the image seems too uniformly coloured, then the image that is normalized will create a greater colour range.

2.5.3 Image Format Analysis

As we mentioned before there are a variety of image file formats that images can be stored in, and these formats contain some information. Sometimes in image forgery, changes to the image modify the file format. In Krawetz [11], as he explained in his paper for JPEG file format, he mentioned that JPEG contains feature sets, and changes to the image will change the feature set. So if the feature set shows manipulations then the manipulations can be identified. The feature set of the JPEG are: metadata, quantization table, etc.

2.5.3.1 Metadata

Image metadata is text information about a picture's pedigree and includes details pertinent to the image itself and other information about its production. These details include camera type (make and model), image resolution, shutter speed value, time the image was taken or altered (timestamps) and other features [11]. There are different types of metadata, some of them are automatically generated by the cameras or devices capturing the image, while others are generated by specific applications. Table 2.1 below shows the common types of metadata.

Common metadata Blocks		
Generated by Cameras	**Generated by Applications**	**Generated by Either**
Maker Notes Print IM	ICC Profile IPTC Photoshop XMP	File EXIF JFIF APP14

Table 2.1: The most common types of metadata

If we take an example, the EXIF (Exchangeable Image File) format in figure 2.5 typically includes lens settings, timestamps, camera models, etc. This metadata can be used to identify information about the camera settings used for the photo.

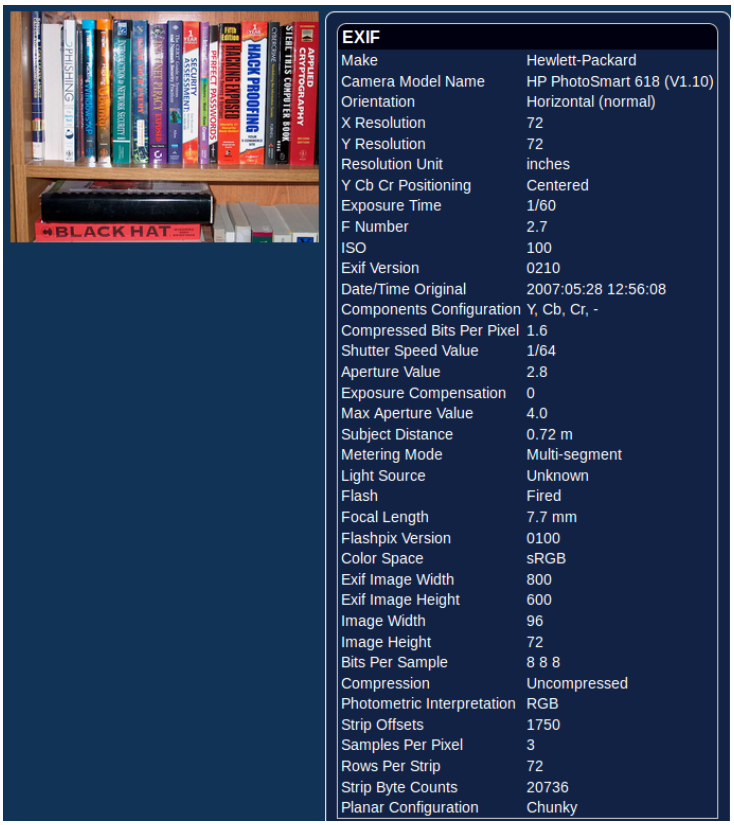


Figure 2.5: The figure show us the camera's make and camera model name (HP PhotoSmart 618), and also the original time this photo was taken 28-May-2007 [7].

As we saw before, the metadata gave us information about how the image file was created and processed. So using that information we can decide if the image was processed by a graphical program, or tampered with to create an illusion. To analyze this metadata information the things you have to consider include:

- **Make, Model and Software:** Tells you the application or device (camera) that created the image.
- **Image size:** The metadata records the dimensions of the image, so you can check if the rendered size of the image (at the bottom of the metadata) match the other

sizes in the metadata, because some applications do not update other metadata fields when they crop or resize images.

- **Timestamps:** As we saw before, timestamps identify time the image was taken or changed, so the question is do the timestamps match the expected time frame?

- **Missing metadata:** We know that images taken by a digital camera should have camera information. Some online services and applications cut metadata. Missing metadata fields usually indicate a resaved image and not an original picture.

Although metadata can give us some good information about a picture, it has a number of limitations:

- Since some of metadata fields are plain text, photo fakers may attempt to edit the metadata, and such edits may not be detectable.

- Misleading metadata information. For example, concatenating two different images using Photoshop, can cause the new image to take metadata of one of the concatenated images. This causes delusive details since the metadata information didn't match the new saved image.

2.5.4 Advanced Image Analysis

The image format analysis we explained above does not evaluate the image itself, it just confirms metadata errors and tools that modified an image. However, techniques like Principal Component Analysis (PCA) and Error Level Analysis (ELA) can identify specific image manipulations.

2.5.4.1 Error Level Analysis

Error Level Analysis (ELA) is a technique that detects changes within compressed images like JPEG, and this helps us to detect or identify fake images. ELA tries to discover if some areas of the image have different compression levels [11]. The way ELA works is by recompressing or resaving JPEG images at a known error level, such as 95%, and then comparing the two images (the original JPEG image and the recompressed image) pixel by pixel by computing the difference between them. With JPEG, unmodified images show the same error level through the whole image, while modified areas of an image have different error levels when it is recompressed. So if a section of the image is at a significantly different error level, then it likely indicates a digital modification. Thus the analysis produced by the ELA could help us identify changes done in the JPEG images [11]. In figure 2.6, the image of the books is manipulated by using Photoshop software, where some of the books in the shelf were duplicated and the dinosaur toy was added. As we see in the right figure, ELA 95 identifies the changes: the previous stable areas in the image are not stable after the modification.

Figure 2.6: Forged image and ELA detected image [7].

2.5.5 Techniques of Clone Detection

2.5.5.1 Exhaustive Search Method

The exhaustive search is a technique to discover duplicate regions in the same image. It compares every pair of regions with each other to identify the copied and pasted regions. Although this technique is the simplest technique for detecting copy paste in an image, the computational time is very high in order to be effective for large images. To reduce the computational cost of this technique, first the image must be segmented into blocks, then the comparison will be by blocks as we will see in section 2.5.5.2 and 2.5.5.3.

2.5.5.2 Block Matching

This method is also good for detecting copy paste forgery by verifying if a set of blocks of pixels in a region of the image matches with another block in the same image. It segments the image of size $(M \times N)$ into $(M-B+1) \times (N-B+1)$ blocks by using $B \times B$ blocks of pixels, which represents the minimum size considered for a match. Then each block is compared with the remaining blocks for matches [23]. Though this technique is a good and useful method for detecting the forged segments in the same image, it has some drawbacks.

- As we know, images sometimes have uniform regions or areas with similar pixel intensities, and so the block matching technique may detect the similar regions in the real image as a duplicate.
- The second problem is the block size, as when the block size is larger than the forged region then there is no exact matching between two blocks and so that means there is no result. Also if the size of the block is smaller, the forged region may cross the boundaries of adjacent blocks and then you would also not find a match. If the size of the block size is made very small, the matching process becomes computationally expensive, specially for large size images.

2.5.5.3 Exact Match

This technique first chooses a block size of say $B \times B$ pixels from the top-left corner, and then moving right and down by striding one pixel at a time along the whole image. Then the pixel values are extracted by columns in each block and placed into a matrix A with B^2 columns and $(M - B + 1) \times (N - B + 1)$ rows [6]. To recognize the similar rows, the rows of the matrix A are sorted using lexicographic sorting and then the similar rows are easily searched by looking at the two successive rows for those that are identical [23]. Below are the detailed steps of exact match technique:

- ☐ If the image is an RGB image, then convert it in to a greyscale image.
- ☐ Divide the greyscale image into a number of overlapping block pixels.
- ☐ For each of the $(M - B + 1) \times (N - B + 1)$ blocks, extract the features, and store the extracted features of each block as rows of a matrix.
- ☐ Sort the matrix in ascending order by using lexicographic sorting.
- ☐ Search successive identical rows and get the respective block positions.

2.5.5.4 Robust Match

The previous matching techniques (block and exact matching) work well only when the duplicate blocks have similar colour intensities (similar greyscale values), but fails if the intensities of the copied part differs from original part because of brightness and contrast adjustments [6]. The best alternative to detect copy-move forgery is Robust Match. The robust match instead of comparing blocks based on their pixel representation compares quantized Discrete Cosine Transform (DCT) coefficient values for each block [6].

The (DCT) coefficients $F(u, v)$ of a given image block $f(x, y)$ of size $N \times N$, can be calculated by the formula:

$$F(u,v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)\alpha(u)\alpha(v) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

where,

$$\alpha(k) = \begin{cases} \sqrt{\frac{1}{N}} & \text{if } k = 0 \\ \sqrt{\frac{2}{N}} & \text{if } k = 1, 2...N-1 \end{cases}$$

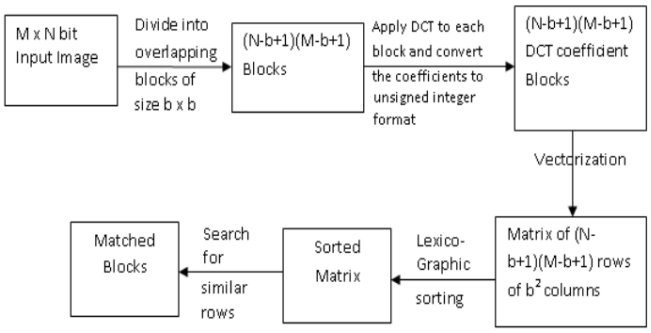


Figure 2.7: Steps of DCT based Robust Detection Method
[6]

After getting the coefficient values, the detection process is the same as in exact matching. The image is scanned by a sliding $B \times B$ block, beginning at the upper left corner until the lower right corner, then it calculates the DCT transform of each block. The DCT coefficients are quantized to get better quality features (good features of image blocks are needed for forgery detection). Then to extract similar rows, the matrix is sorted using lexicographic sorting and then the quantized values of the DCT coefficients of each block are compared. If there is a matching block then the algorithm stores the positions of the matching blocks and increments a shift-vector counter called C. Formally, let(i_1, i_2) and (j_1, j_2) be the positions of the two matching blocks [23]. The shift vector s between the two matching blocks is calculated as:

$$s = (s_1, s_2) = (i_1 - j_1, i_2 - j_2)$$

Then the shift vector counter C is incremented by 1 (shift vector counter value starts at zero and is increased whenever a block match occurs). At the end, the counter C indicates the frequencies with which different shift vectors occur. Then the shift vectors $s_1, s_2, ..., s_k$, whose occurrence exceeds a user-specified threshold T that is $C(s_r) > T$ for $r = 1, ..., k$, are determined. For all such shift vectors, the matching blocks that contributed are identified as segments that might have been copied and moved. Note that in the Robust and Exact Matching techniques, if the image is a colour image, then one must first convert to a greyscale image by using the formula $I = 0.299R + 0.587G + 0.114B$ [23].

The Robust Match algorithm steps are :

- Gray-scale conversion: convert a colour image into greyscale.
- Forming blocks: Divide the greyscale image into overlapping blocks.
- Extracting features: For each of the $(M - b + 1) * (N - b + 1)$ blocks, compute the DCT coefficient values and quantize them. Store the quantized coefficients for the blocks as the rows of a matrix. Also store the co-ordinates of the top-left corner pixel (x, y) as the location of the block.

- Lexicographic sorting: Sort the matrix in ascending order.
- Matching process: Search for consecutive identical rows and if there is a match store the positions of both the blocks in a separate list.
- Compute the shift vectors: Compute shift vectors for every pair of matching blocks using equation: $s = (s_1, s_2) = |(i_1 - j_1, i_2 - j_2)|$ and set the value of a shift vector counter $c(s_1, s_2)$ to 1. If the shift vector value is a repetition, increment its counter value by 1.
- Mark the blocks: Find all shift vectors whose counter values exceed a user specified threshold value T. Mark all these blocks with some colour to indicate duplication. If there are no such shift vectors, declare that the image is not tampered.

2.6 Conclusion

In this chapter, we introduced some information about digital photographs and their creation pipeline, image formats and how files are compressed. We also introduced some techniques used to forge the digital images and their types, and also some techniques and methods surrounding image forgery detection. In chapter 3, we will introduce research methodology that outlines some important tools that are implemented to detect image forgery and convolutional neural network algorithm that we will implement to classify fake images from real images.

Chapter 3

Research Methodology

3.1 Introduction

In the previous chapter we discussed the research and related work surrounding fake image detection. This chapter will outline some important tools that are implemented to detect image forgery, and using these tools we want to check how well these tools can detect faked or tampered images. We will also outline convolutional neural network algorithm that we can train with CASIA 2.0 dataset to classify fake images from authentic images.

Section 3.2 describes the research problems. Section 3.3 describes the implemented tools that we use to check our tampered images, and to check how well these tools can detect faked or tampered images. Section 3.4 describes CASIA 2.0 dataset of images that we will use to train our machine learning algorithms. Section 3.5 describes convolutional neural network algorithm that we will implement to classify the images into authentic and fake.

3.2 Research Problem

The purpose of this work is to design and build a convolutional neural network (CNN) with the aim of detecting forgeries in images. We will train and test it on the Casia 2.0 image dataset.
The network classifies images into:

1. Real and fake using two class classification method.

2. Real, copy-paste and spliced images using three class classification, and both stages we will print the accuracy of the network after training and testing.

Finally we will compare this cascade of CNN to the one of the implemented tools, Forensically, and check how well both are classifying images into real and fake.

3.3 Implemented Tools

3.3.1 Forensically

Forensically is a set of free tools for digital image forensics made by Jonas Wagner [27]. It allows you to discover details and manipulations in digital images. The free Forensically tool includes magnifier, error level analysis, clone detection, and metadata extraction.

1. **Magnifier:** The magnifier allows you to see small hidden details in an image by increasing the size of pixels and magnifying the contrast within the window.

2. **Clone Detection:** Clone detector highlights regions that have been copied across using the clone tool copy paste or something like it. It means they detect similar regions within an image. Clone detection technique has quite a few parameters that are:

   Minimal Similarity: Determines how similar two regions need to be in order to be considered clones or how similar the cloned regions need to be to the original.

   Minimal Detail: Controls how much detail an area needs to be considered for the clone detection. So blocks with less detail than this are not considered when searching for clones.

   Minimal Cluster Size: Determines how many clones of a similar region need to be found in order for them to show up as results. Or in other words, it determines how many blocks regions need to share in common in order to be considered a clone.

   Block-size: Determines size of the blocks that are used for the clone detection.

3. **Error Level Analysis:**
   This tool first compresses the image and then compares the original image to a re-compressed version. This tool can make tampered regions brighter than regions which have not been manipulated.
   JPEG Quality: This should match the original quality of the image that has been photo-shopped.
   Error Scale: Makes the differences between the original and the re-compressed image bigger.

3.3.2 Ghiro

Ghiro is a digital image forensics tool that is designed to process and analyze a massive amount of images [28]. It extracts information from provided images and displays them in a nicely formatted report. It also supports some image file types like gif, png, jpeg, bmp, etc. This tool can be used in many real life situations. Forensic investigators could use it on a daily basis in their analysis lab but also people interested to uncover secrets hidden in images could benefit. This software is designed to support large numbers of images and is useful if you need to extract all data and meta-data hidden in an image in

a fully automated way. It is also useful if you need to analyze a lot of images and you have not much time to read the report for all of them.

3.4 Data Preparation

CASIA 2.0 is a collection of natural colour images and manipulated images. This dataset was created by Jing Dong for research on the detection of image manipulation [8]. The dataset contains two folders: the first is Au that contains 7491 authentic images and the second is Tp that contains 5123 images from Au altered using manipulation methods. These images have different sizes, between 240×160 and 900×600 pixels. The formats of the files are JPEG, BMP and TIFF [8].

Brief Descriptions on Design Principles:

Adobe Photoshop CS3 was used when generating tampered images. Tampered regions are either from the same real image or from another image. The following main points are underlined during the dataset preparation.

Content diversity: The real images are images from nature and categorized into nine parts (architecture, character, animals, article, scene, plant, nature, indoor and texture).

Realistic operation: It simulated the process of tampering within the following different ways: Randomly crop-and-paste image region(s) and then the cropped image region can be processed with rotation, re-sizing or other distortion, then pasted to generate a spliced image; different sizes of spliced regions are concerned; most spliced images are prepared to be realistic images judged by human observation [8]. Figure 3.1 show us examples of the dataset.

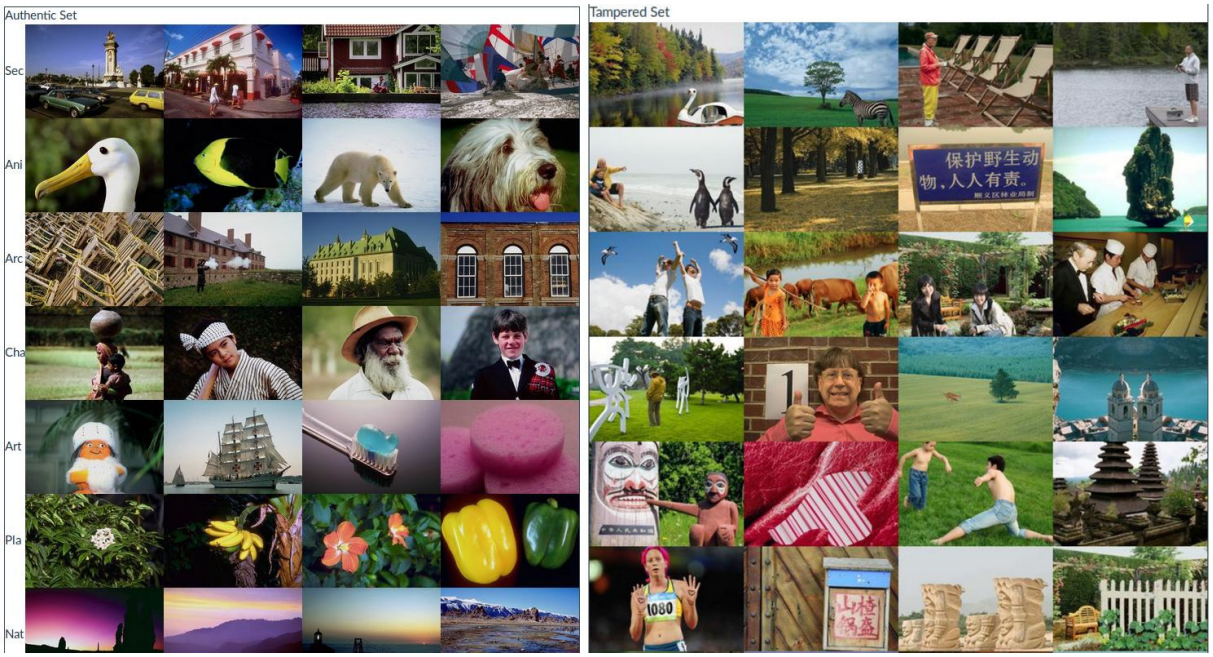


Figure 3.1: Left: authentic set of the dataset. Right: tampered set [8].

3.5 Convolutional Neural Network (CNN)

Convolutional neural networks (CNN) are inspired by the workings of visual cortices in the brains of living organisms that posses sight and are capable of image processing and recognition. In 1980s Yann LeCun designed and trained CNN [29].

3.5.1 Convolution

Convolution is a mathematical operation which describes a rule of how to mix two functions: (1) The feature map (or input data) and (2) the convolution kernel, to form a transformed feature map.
So if we assume x and w are two functions on $A \subseteq \mathbb{R}$ then the convolution of x with w is given by:

$$(x * w)(t) := \int_A x(a)w(t-a)da.$$

In the case $A = \mathbb{R}$ then,

$$(x * w)(t) = \int_{-\infty}^{\infty} x(a)w(t-a)da.$$

In image processing, we deal with discrete convolution with 2-dimensions which can be defined as:

$$s[i,j] = \sum_m \sum_n I[m,n]K[i-m, j-n].$$

3.5.2 Convolutional neural network

Convolutional neural network(CNN), is a sequence of convolution layers with non-linear activation functions. That means that like artificial neural networks (ANN), CNN consist of neurons that get inputs and weights, then perform dot product between the inputs and weights, and follow with a non-linear function. But the difference between ANN and CNN is that in ANN each neuron in the hidden layer is fully connected to all neurons in the previous layer. This fully connected layer works fine if the image is small, for example $32 \times 32 \times 3 = 3072$. So a single fully connected neuron in the hidden layer of a ANN has 3072 parameters. These neurons in the hidden layer can manage this number of parameters. But using larger scale images that have e.g. $120,000$ parameters, the fully connected layer cannot manage these large numbers of parameters. This large number of parameters will lead to parameter overfitting. On the other hand, CNN connects each neuron in the hidden layer to a small region of the input image called the receptive field. So CNN have far fewer connections and parameters than ANN, and also CNN are easier to train [30].

The reason why we are using CNN is that CNN architecture implicitly combines the benefits obtained by a standard neural network training with the convolution operation

to efficiently classify images. Further, being a neural network, the CNN (and its variants) are also scalable for large datasets, which is often the case when images are to be classified. A CNN takes just the image's raw pixel data as input and "learns" how to extract higher- and higher-level representations of the image content, and ultimately infer what object they constitute.

3.5.3 Convolutional neural network architecture

CNN consists of a set of layers. The layers are the main building blocks in the network architecture that have interconnected neurons. Layers receive weighted inputs, and transforms them with a non-linear function. It then passes the output to the next layers. There are four main layers in convolutional network architecture: convolution layer, pooling layer, fully connected layer and ReLU Layer (Rectified Linear Units).

1. **Convolution layer:** The convolution layer is the main part of the CNN that usually contains several feature maps. Parameters of the convolution layer are learnable filters. During convolving the filter through all positions of the input computing dot product between the filter values and the input, the network will learn filters that activate when they see some specific type of features at some special position on the input. The output volume of the convolution layer is an array of neurons arranged in a 3 dimensional volume. So the convolution layer transforms 3D input (image) into 3D output [9].

2. **Pooling layer:** Pooling layer is the part of the convolutional network architectures that is applied after the convolution layers. Its function is to take the convolution layer output and reduce its dimensionality. That means it takes input over a certain area and reduces it to a single value. So the pooling layer reduces the number of parameters needed in later layers, and the number of computations in the network, and also prevents parameter overfitting by controlling the parameters. When the area of the pooling layer is large, a lot of information is thrown away, so performance of prediction will decrease. The common technique for pooling is max pooling as we see in figure 3.2 right, the maximum value of the four values is selected. There is also L2 pooling which takes the square root of the sum of squares of the activations in a 2×2 region [30].

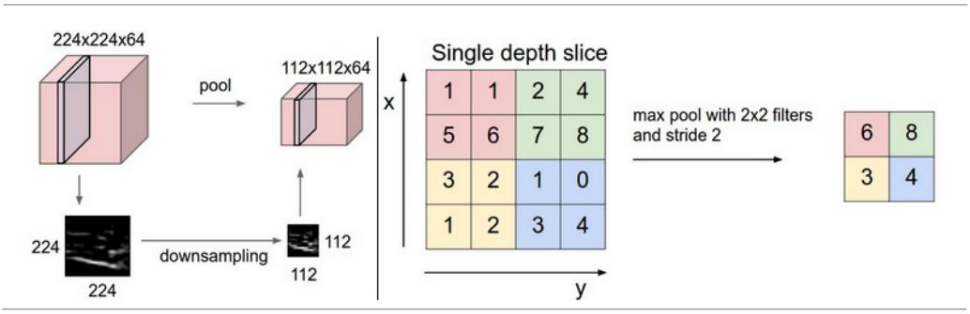


Figure 3.2: Left: Input volume that have $224 \times 224 \times 64$ size, after pooling with filter of size 2×2 and stride 2 the output volume size became $112 \times 112 \times 64$ with the same depth of original input. Right: Max pooling that have 2×2 filter and stride 2 [9].

Pooling helps convolutional neural networks to use more convolutional layers because it reduces memory consumption and so allows the network to use more convolutional layers or more feature maps.

3. **Fully Connected Layer:** A fully connected layer is the last layer of the convolutional architecture. It comes after the convolution and pooling layers. A fully connected layer contains neurons that connect to the entire input volume, as in ordinary neural networks i.e. each neuron in the hidden layer is connected to all neurons in the previous layer [9].

4. **ReLU Layer:** There is a also another layer called ReLU (Rectified Linear Units) that is used instead of a linear activation function to add non-linearity to the network, otherwise the network would only ever be able to compute a linear function.

3.5.4 Putting all together

To get a better understanding how the convolutional network layers are connected and how they work together, let us take as an example of a $LeNet5$ network which is designed for handwritten digits and works well on the digit classification task. It consist of a convolution layer followed by pooling layer, and then another convolution layer followed by pooling layer and finally fully connected layers. These layers make an architecture of [Conv-Pool-Conv-Pool-Conv-FC], as we see in figure 3.3.

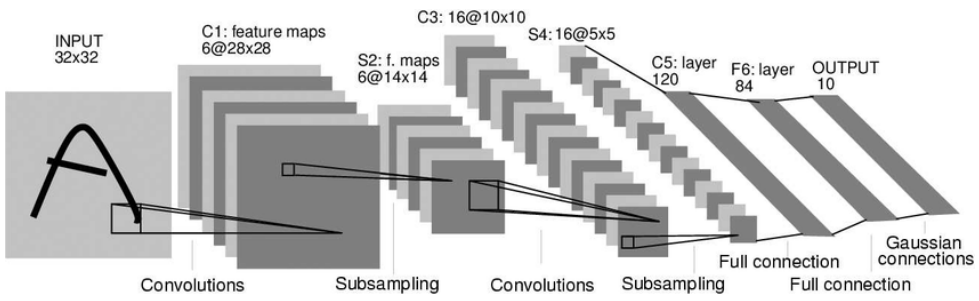


Figure 3.3: Architecture of LeNet-5 [10].

To explain the figure, the network begins with input image of 32×32 size with one channel (i.e grayscale). Layer C_1 is a convolution layer with 6 different feature maps of size 28×28. Each feature map has a 5×5 receptive field in the input layer. Layer S_2 is a subsampling layer that reduces the dimension of the image using max pooling. This is applied to a 2×2 region across each of the 6 feature maps. The result is a layer of $6 \times 14 \times 14$ hidden feature neurons. Layer F_6 is a fully connected unit that connects the output layer.

Using CNN, we want to build an image classification program and giving it a dataset of images (Casia 2.0 dataset) then it will determine image forgeries and authenticity. The network first classifies images into real and fake using two class classification. Secondly it classifies images into real copy-paste and spliced images using three class classification, and both stages we will print the accuracy of the network after training and testing. Finally we will compare this convolutional neural network to one of the implemented tools, Forensically, and check how well both are classifying images into real and fake and type of fake.

Chapter 4

Results And Discussion

4.1 Convolutional neural network architecture

To train our network first we have to build the Convolutional Neural Network, that is composed of the following layers: Three convolutional layers, a rectified linear unit (ReLU) layer that introduces non-linearity, two pooling layers that allow us to reduce the size of the input and make feature detection more robust by making it more impervious to scale and orientation changes, the stride of pooling layer in this experiment is one. Flatten layer that is a simple layer that takes data from pooling layer or sometimes when there is no pooling layer it takes data from convolution layer and converts into a 1D feature vector to be the input of the final layer. Fully connected layer is the final layer that takes inputs from the previous layer (Flatten layer) and connects it to every single neuron it has. There is also a drop out layer between some layers. The drop out layer prevents parameter over-fitting. It acts as a way of regularization or it is a regularization method that stochastically sets to zero the activations of hidden units for each training case at training time, and the reason is that the number parameters in the layers are huge and then it is likely to overfit [31].

We also use some open source libraries like Keras. Keras is a very famous and easy deep learning library written in Python and it focuses on enabling fast experimentation [32]. It has several backends like; Theano, TensorFlow, CNTK backends, in this project we are using Keras with TensorFlow backends. Keras will run much faster with an Nvidia GPU. Figure 4.1 is the architecture of the our CNN as produced by the keras program.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 32, 50, 50)	896
activation_1 (Activation)	(None, 32, 50, 50)	0
conv2d_2 (Conv2D)	(None, 32, 48, 48)	9248
activation_2 (Activation)	(None, 32, 48, 48)	0
max_pooling2d_1 (MaxPooling2	(None, 32, 24, 24)	0
conv2d_3 (Conv2D)	(None, 64, 22, 22)	18496
activation_3 (Activation)	(None, 64, 22, 22)	0
max_pooling2d_2 (MaxPooling2	(None, 64, 11, 11)	0
flatten_1 (Flatten)	(None, 7744)	0
dense_1 (Dense)	(None, 64)	495680
activation_4 (Activation)	(None, 64)	0
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 2)	130
activation_5 (Activation)	(None, 2)	0

Figure 4.1: Convolutional neural network architecture

4.2 Training and results

Before we trained the network, we first preprocessed the images by resizing the images into 50×50 pixel width and height. Then we serialized into an array that contains $50 \times 50 \times 3 = 7500$ values - since each pixel has three components RGB(red, green, blue), we multiplied by three. In training, as the network runs it returns the training loss (number of images mislabelled) and training accuracy (number of images correctly classified) of the training set, and also validation loss and validation accuracy (number of images correctly classified) of the validation set. Since we are using 2 and 3 class classification problems, we will use "adadelta" optimizer and also "categorical cross entropy" loss to train the convolution network.

1. **Real/fake image classification**

   In two class classification, there are two output neurons in the network, the first one representing the real image and the second neuron representing the fake image. For example, after running 20 epochs the network gave us the results in Table 4.1:

Epoch	Train loss	Valid loss	Train acc	Valid acc
1	0.6039	0.5525	0.6705	0.7065
2	0.5235	0.5213	0.7401	0.7428
3	0.4973	0.4818	0.7628	0.7701
4	0.4674	0.4649	0.7798	0.7697
5	0.4337	0.4500	0.7967	0.7882
6	0.390	0.4314	0.8288	0.8048
7	0.3477	0.4582	0.8511	0.7882
8	0.3067	0.4450	0.8728	0.8168
9	0.2712	0.4146	0.8867	0.8221
10	0.2333	0.4147	0.9041	0.8382
⋮	⋮	⋮	⋮	⋮
18	0.1174	0.5287	0.9589	0.8611
19	0.1085	0.5337	0.9638	0.8458
20	0.1053	0.4881	0.9644	0.8563

Table 4.1: Convolutonal neural network result of train loss/valid loss and train accuracy/valid accuracy

We trained convolutional neural network with CASIA dataset, from the dataset we have used 7950 authentic and fake images for training, the remaining data we used for testing of the network. As we can see in the table 4.1 the accuracy of the convolutional neural network at the end is 85.63%. We observe a very good performance even though we have used 20 epochs of training. That means the trained CNN network is able to recognize the image as real or fake at a maximum success rate of 85.63%. So the use of this application can be used as a false proof technique in digital authentication. It could also greatly reduce spreading of fake images through websites and through social media.

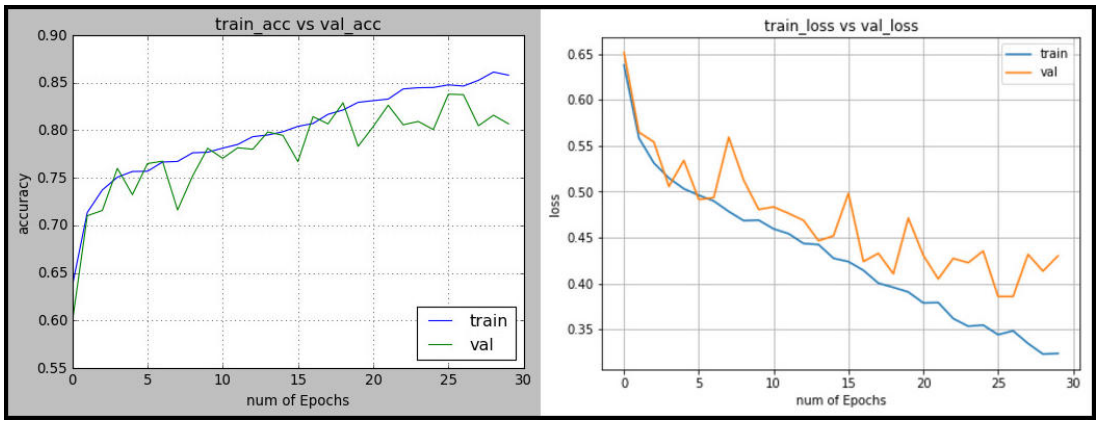


Figure 4.2: Left: training accuracy vs validation accuracy. Right: training loss vs validation loss.

Figure 4.2 right shows the error (loss) for the training and validation set. As the

number of epochs increases the error decreases, meaning that the network is learning well. Figure 4.2 left shows that as the number of iterations (epochs) increases the accuracy increases for both training and validation meaning that the network generalized well to data that the network has not seen before.

We use a confusion matrix that describes the performance of a classification model. It give us a detailed representation of what is going on with the labels. The diagonal elements in the confusion matrix represent the number of points for which the predicted label is equal to the true label, while off-diagonal elements are those that are mislabelled by the classifier. As we see in figure 4.3, the diagonal is where the classification is more dense, showing the good performance of our classifier. In our CNN we used 2484 images to test the network (test dataset), where 1440 of the test dataset are real images and 1044 are fake images. In class(0) of the first block of the confusion showing us that 1244 of real image were correctly classified as real, and 196 real images were classified as a fake. In class(1), the fake images block, 889 fake images were correctly classified as a fake, while 155 fake images were classified as a real.

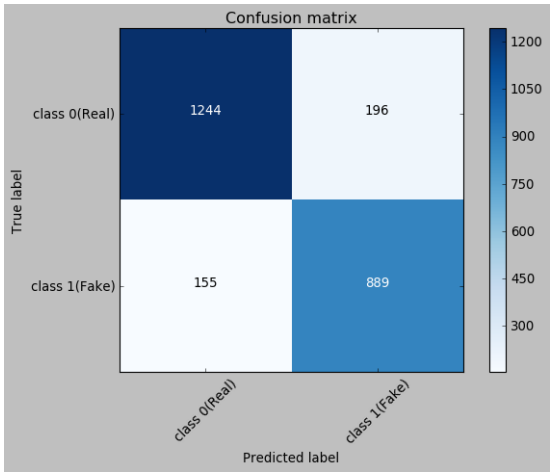


Figure 4.3: Confusion matrix for two classes

	Precision	**Recall**	**F1-score**	**Support**
Class 0(Real)	0.89	0.86	0.88	1440
Class 1(Fake)	0.82	0.85	0.84	1044
Avg / Total	0.86	0.86	0.86	2484

Table 4.2

In table 4.2 we also used precision and recall that are determinable from the classification report. Recall gave us individual class accuracy where real image class accuracy is 86% and fake image class accuracy is 85%. $F1$ score is a aggregation of the two (precision and score) of each class, telling us that the real image class are

classified well at accuracy of 88% better than the fake image class that has 84%. Support column contains the total number of actual real and fake images of the test dataset.

2. **Real/spliced/copy-paste classification**

   For three class classification there are three output neurons; real image, spliced image and copy-paste image. i.e. we use the same architecture except for three output classes and trained on the same data set. This time we used same network configuration but we divided the CASIA dataset into three parts (real, copy-paste, and spliced). And we used confusion matrix but this time the classification is very poor as we see in figure 4.4.

   - The real images in class (0) were classified nicely, where 1183 of real images where classified as a real and 38 of the real images were miss-classified as a spliced image and 219 real images were miss-classified as a copy-paste images.
   - The spliced images in class(1) were classified very badly, where only 54 of the spliced images where correctly classified as a spliced images, 41 of them were miss-classified as real and 280 of them were miss-classified as copy-paste images.
   - The copy-paste images in class(2) were classified not as badly, where 481 of images were correctly classified as a copy-paste images, 113 of them were miss-classified as spliced and 75 of them were miss-classified as real images.

   The reason why this second class is poorly classified is the CASIA dataset is a small dataset and in particularly the number of spliced images in the dataset is smaller than the other real and copy-paste images in the dataset. So to divide that small dataset into three classes and then each class has its data divided into training validation and testing cause poor performance of the network. That is the reason why the second class (spliced) is completely lost.

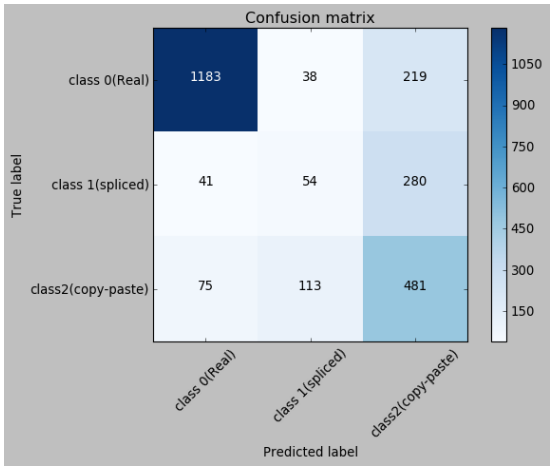


Figure 4.4: Confusion matrix for three classes

In table 4.3 we also used precision and recall, as before but now in three classes. Real image class accuracy is 82%, spliced image class accuracy is 14% and copy-paste image class accuracy is 72%. The overall network accuracy for three classes is 69%.

	Precision	**Recall**	**F1-score**	**Support**
Class 0(Real)	0.91	0.82	0.86	1440
Class 1(spliced)	0.26	0.14	0.19	375
class 3(copy-paste)	0.49	0.72	0.58	669
Avg / Total	0.70	0.69	0.69	2484

Table 4.3

In general CNN often require a large amount of training data, and even with high-tech computers with GPUs and plenty of computational resources can take many hours to train.

However, overall, this work suggests that convolutional neural network or deep learning in general is a promising approach for image forensics (as it is for many other tasks in computer vision). These results may not give you high accuracy, but they are promising and can hopefully encourage further exploration in this direction.

3. **Comparing convolutional neural network and Forensically tool**

   As we mentioned before in chapter 3 section 3.3, Forensically is a digital image forensic tool. It allows us to discover details and manipulations in digital images. It can discover cloned and spliced images by highlighting the cloned regions and spliced images. To compare CNN and Forensically, figure 4.5 consists of 10 images divided into three rows. The first row consist of 4 real images, the second row consist of 3 spliced images and the third row consist of 3 copy-pasted images. The classification of the images are in table 4.4.

Figure 4.5: Some real and fake images

Images in rows	**Forensically**	**CNN**
Real images	All 4 were classified as real	All 4 were classified as real
Spliced images	All were classified as spliced	All were miss-classified
Copy-paste images	All were classified as copy-paste images	2 were classified as copy-paste

Table 4.4: Comparing Forensically and CNN

Figure 4.6 shows us how Forensically analyzes copy-paste images. The first row is the input image and the second row is the visualization of the image after applying Forensically.

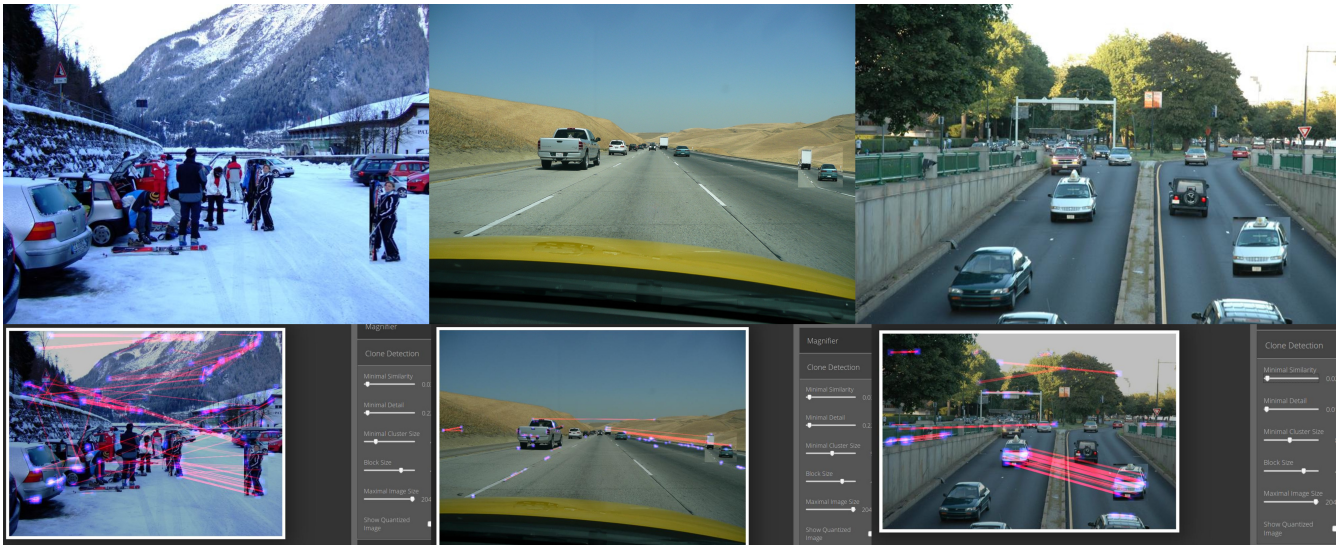


Figure 4.6: First row: the images before Forensically is applied. Second row: after Forensically is applied highlighting the copy-pasted regions.

Figure 4.7 shows us how Forensically analyzed spliced images, where the first row is the input image, and the second row is the visualization of the image after applying Forensically that highlighted parts of the image that have different compression levels. The first image shows us that the image of the bird was added to the image, because it has a different compression level. The second image shows us that the two animals in the corner are added to the image. The third image also shows us that the upper image was added.

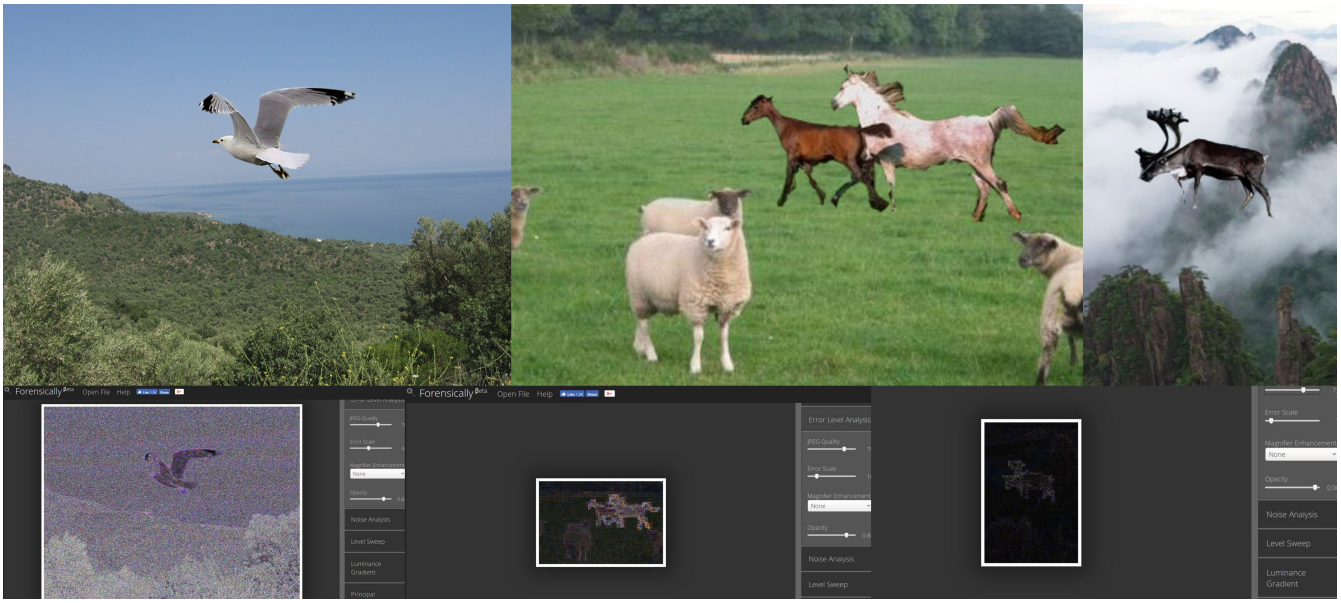


Figure 4.7: First row: the images before Forensically. Second row: after Forensically highlighting the spliced parts.

Chapter 5

Conclusion

With the cheap and sophisticated image processing technologies and editing software now available, image forgeries are rising. An enormous amount of fake images are made by editing software in practice. Therefore, identification techniques for faked images are regarded as an important research field. In this project I have given a brief idea about image forgery and its detection methods and I have mentioned some important image authentication tools in daily life for investigators in image forensics. I outlined also an idea about image forgery detection using CNN that is a machine learning classifier model. I trained the CNN by using the CASIA 2.0 dataset and was able to successfully distinguish between real and faked images in general.

In chapter 2, we introduced some information about digital photographs and their creation pipeline, image formats and how files are compressed. We also introduced some techniques used to forge the digital images and their types, and also some techniques and methods surrounding image forgery detection. Chapter 3, introduced research methodology that outlines some important tools that are implemented to detect image forgery and CNN that can classify fake images from real images. In chapter 4 we introduced the architecture of the CNN that I built and how I trained the CNN to classify the images.The results obtained after training the CNN are presented. The network first classifies the images into real and fake at the accuracy of 85.63%. That means the trained CNN network is able to recognize the image as real or fake at a maximum success rate of 85.63%. So the use of this application can be used as a false proof technique in digital authentication. It will also greatly reduce spreading of fake images through websites and through social media. Second, the CNN network classifies the images into three classes: real, spliced and copy-paste classes, but the classification is very poor comparing to the two class classification and gave me accuracy of 69%. Finally we compared the implemented CNN network and the Forensically tools that were implemented before.

In general CNN often require large amounts of training data, and even with high-tech computers with GPUs and plenty of computational resources can take hours to train. However, overall, convolutional neural network or deep learning in general is a

promising approach for image forensics (as it is for many other tasks in computer vision). These results may not give you high accuracy, but they are promising and can hopefully encourage further exploration in this direction.

References

[1] S. D. Anderson, *Digital image analysis: Analytical framework for authenticating digital images.* MSC thesis,University of Colorado at Denver, 2011.

[2] M. K. Johnson and H. Farid, "Exposing digital forgeries by detecting inconsistencies in lighting," in *Proceedings of the 7th workshop on Multimedia and security*, pp. 1–3, ACM, 2005.

[3] Y. Q. Shi, C. Chen, and W. Chen, "A natural image model approach to splicing detection," in *Proceedings of the 9th workshop on Multimedia & security*, pp. 1–3, ACM, 2007.

[4] H. Farid, "Digital doctoring:can we trust photographs?," *http://www.cs.dartmouth.edu/farid/downloads/publications/deception09.pdf*, pp. 3–5, 2003.

[5] J. Y.-F. Hsu, *Image tampering detection for forensics applications.* PhD thesis, Columbia University, 2009.

[6] M. Mishra and M. Adhikary, "Detection of clones in digital images," *arXiv preprint arXiv:1407.6879*, pp. 91–99, 2014.

[7] H. Factor, "Fotoforensics," *http://Fotoforensics.com/tutorial-meta.php*, 2012-2017.

[8] J. Dong and W. Wang, "Casia tampered image detection evaluation database," *http://forensics.idealtest.org/*, 2011. Downloaded 3-February-2017.

[9] A. Karpathy, "Convolutional Neural Networks.," *URL:http://cs231n. github. io/convolutional-networks*, 2015.

[10] V. Ayumi, L. R. Rere, M. I. Fanany, and A. M. Arymurthy, "Optimization of convolutional neural network using microcanonical annealing algorithm," in *Advanced Computer Science and Information Systems (ICACSIS), 2016 International Conference on*, pp. 506–511, IEEE, 2016.

[11] N. Krawetz and H. F. Solutions, "A picture's worth," *Black Hat Briefings USA*, 2007.

[12] J. R. Sturak, "Forensic analysis of digital image tampering," tech. rep., Air force institute of technical Wright-Patterson school of engineering and management, 2004.

[13] H. Farid, "Digital doctoring: how to tell the real from the fake," *Significance*, vol. 3, no. 4, pp. 2–4, 2006.

[14] D. P. Curtin, "Sensors, pixels and image sizes," *An Extension to PhotoCourse Digital Photography Textbooks*, pp. 11–12, 2011.

[15] Wikipedia, "Telephoto lens wikipedia, the free encyclopedia," 2017. [Online; accessed 3-August-2017].

[16] T. Moynihan, "CMOS is winning the camera sensor battle, and here's why," *TechHive.[online]*, 2011.

[17] R. Lukac and K. N. Plataniotis, "Color filter arrays: Design and performance analysis," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 4, pp. 1260–1267, 2005.

[18] R. Kimmel, "Demosaicing: image reconstruction from color CCD samples," *IEEE Transactions on image processing*, vol. 8, no. 9, p. 1221, 1999.

[19] M. Singh, S. Singh, "Various image compression techniques: Lossy and lossless," *Image*, vol. 142, no. 6, pp. 23–26, 2016.

[20] L. K. Tan, "Image file formats," *Biomedical imaging and intervention journal*, vol. 2, no. 1, pp. 1–4, 2006.

[21] Y. Parashar and S. M. Dubey, "Survey of digital image tampering techniques," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 2, pp. 415–420, 2016.

[22] V. P. Raval, "Analysis and detection of image forgery methodologies," *International Journal of Scientific Research and Development*, vol. 1, no. 9, pp. 415–418, 2013.

[23] A. J. Fridrich, B. D. Soukal, and A. J. Lukáš, "Detection of copy-move forgery in digital images," in *Proceedings of Digital Forensic Research Workshop*, Citeseer, 2003.

[24] M. Nizza and P. J. Lyons, "In an Iranian image, a missile too many," *The Lede, The New York Times News Blog*, 2008.

[25] D. Sharma and P. Abrol, "Digital image tampering a threat to security management," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 10, pp. 4120–4121, 2013.

[26] S. G. Rasse, "Review of detection of digital image splicing forgeries with illumination color estimation," *International Journal of Emerging Research in Management and Technology*, vol. 3, pp. 27–28, 2014.

[27] W. Jonas, "Photo-forensics," *https://29a.ch/photo-forensics/help*, 2015.

[28] M. B. Alessandro Tanasi, "Forensic analysis," *https://media.readthedocs.org/pdf/ghiro/ghiro-0.1/ghiro.pdf*, pp. 3–4, 2014.

[29] Y. Bengio *et al.*, "Learning deep architectures for AI," *Foundations and trends in Machine Learning*, vol. 2, no. 1, pp. 24–25, 2009.

[30] M. A. Nielsen, "Neural networks and deep learning/online book," 2015.

[31] H. Wu and X. Gu, "Towards dropout training for convolutional neural networks," *Neural Networks*, vol. 71, pp. 1–10, 2015.

[32] S. Persson, "Application of the german traffic sign recognition benchmark on the vgg16 network using transfer learning and bottleneck features in keras, www.diva-portal.org/smash/record.jsf?pid=diva2:1188243," 2018.