TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Huu Phuc Nguyen 184663IVCM

RESEARCH METHOD IN DETECTING SWAPPED FACE IMAGE AND VIDEO FORGERY

Master's Thesis

Supervisor: Matthew James Sorell
Prof

Tallinn 2019

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Huu Phuc Nguyen 184663IVCM

UURIMISMEETOD TUVASTAMAKS NÄO VAHETUST VÕLTSINGUT PILDIST JA VIDEOST

Magistritöö

Juhendaja: Matthew James Sorell

Prof

Tallinn 2019

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis and this thesis has not been presented for examination or submitted for defence anywhere else. All the used materials, references to the literature and the work of others have been cited.

Author: Huu Phuc Nguyen

27.10.2019

Abstract

The video forgery has been becoming an emergent attack vector. Detecting fake video that has been facing with cumbersome has paid attention by general researchers. Manipulated video affects directly to the worthy trust of video evidence because the video can be valuable evidence in court. Furthermore, fake video that is able to degrade dignity and morality of individuals or organizations can be propagated through Internet straightforwadly. This is the reason the government and organizations channels into developing method for countermeasure of falsified video. Our thesis means to contribute to solve this problem, objective of thesis is to use image processing and signal processing technique to detect the fake video. Our strategy is that we pay efforts to observe the aperture of the fake video. As a result, detection method will be developed to expose the forgery and we purpose to solve the limitation of existing deepfake detection method.

The thesis is in English and contains 54 pages of text, 6 chapters, 31 figures, 11 tables.

Annotatsioon

Uurimismeetod tuvastamaks näo vahetust võltsingut pildist ja videost

Video võltsimine on muutumas esilekerkivaks rünnakuvektoriks. Videovõltsingu tuvastamine, mis on tekitanud palju tüli, on teadlaste tähelepanu pööranud. Manipuleeritud video mõjutab otseselt videotõendite usaldust, kuna videod võivad olla kohtus väärtuslikud tõendid. Lisaks sellele saab Interneti teel levitada võltsvideot, mis võib alandada inimeste või organisatsioonide väärikust ja moraali. See ongi põhjuseks, miks valitsus ja organisatsioonid üritavad välja töötada vastumeetmeid võltsvideotele. Meie lõputöö eesmärgiks on aidata leida lahendus sellele probleemile ja töös kasutame võltsitud video tuvastamiseks pildi- ja signaalitöötlust. Meie strateegiaks on tähelepanu suunamine filmitud võltsvideo kaamera avale. Selle tulemusel töötame välja võltsimise tuvastusmeetodi ja meie eesmärk on lahendada juba olemasolevad sügava pinna tuvastamise meetodi piirangud.

Lõputöö on kirjutatud Inglise keeles ning sisaldab teksti 54 leheküljel, 6 peatükki, 31 Figuret, 11 tabelit.

List of abbreviations and terms

AI	Artificial Intelligence
HOGs	Histogram of Oriented Gradients
GANs	Generative Adversarial Networks
CNN	Convolutional Neural Network
RGB	Red, Green, Blue
ML	Machine Learning
DIP	Digital Image Processing

Tables of contents

Lists of figures

Lists of tables

1 Introduction

In this chapter, we are going to discuss about the introduction of thesis. In the section 1.1, background knowledge of thesis is mentioned. In the next section, we indicate the problem statement of thesis and then we discuss about objects and scopes of thesis. In the last section, thesis structure is modelled.

1.1 Background and Motivation

In recent years, artificial intelligence (AI) has drawn attention by general public and especially, by the scientific researchers because of its ubiquitous application. Machine learning is a branch of AI that enable computer system to learn from images, video, examples. Not only does AI continuously improve the performance by learning from experience as human beings do but also adjust behaviour based on prior performance and new inputs. The first-generation AI technology has been published such as Apple's Siri, Amazon's Alexa. This kind of software can identify queries and requests using spoken human language and responds using answers from database.

There is an emergent attack trend that using Artificial Intelligence or Machine Learning (ML) to merge, combine, replace and superimpose images and video clips onto a video. That creates a fake video looks authentic, this product is named deepfake videos. The attempt to swap somebody's face with the other's face happened long time ago. However, with the support of AI, fake videos become prominent and straightforward to many people.

Thanks to the development of technology, an area of sciences researches on digital image forensics that is developing in leap and bound to detect image forgeries. However, this is a cat and mouse game because if a new method of detection is designed, the attackers can fight against the method. This is a wide topic area research field because applying modern hardware or software or both of them can improve the result. On the one hand, some kinds of GPU integrate

in computer that has great contribution in the problem. On the other hand, the software is ever changing to improve the speed and reduce complexity, especially multiprogramming [1] is a basic principle of parallel processing and powerful technique to apply for complex algorithm.

For these reasons, many researchers have proposed methods for detecting deepfake videos. We will discuss about their method in detail later. In addition, we would like to contribute to video forensic research by developing an algorithm that can expose falsified videos.

1.2 Problem Statement

At first, it is important to recognize that deepfake videos do not have any ethical issues if it does not cause damage to anyone. Nevertheless, many people take advantage of it to serve as attack apparatus for personal profit. If someone use image of another person without permission and create fake videos to degrade the fame or criticize this person, this action will be considered as human right violation and committed crime. There are existing smartphone and desktop application to generate the image and video forgeries such as FaceApp [2] and FakeApp [3] that are built on Machine Learning technology. These applications do not require users to have scientific knowledge to use. The problem is that with deepfake video everyone can create the false content without the authentication. Deepfake video can swap person face onto another in an image or a video. Image and recorded video have been playing an instrumental role in propagating news through media in every country. Every single day, there are almost 5 billion videos that are watched on YouTube [4]. Day by day, the number of videos that are uploaded in Internet is increasing, however, among them, there are unauthentic videos that are manipulated, which means to distort the truth. Specifically, the algorithm can create the fake video and then improve by continuing to mimic the facial expressions, gestures, voice and make itself become more realistic. Consequently, when the data of a person such as video, audio, image is sufficient, the algorithm can make the person say things that they do not say actually. And this video might not be distinguished by human vision. This exert adverse on the truth of video evidence through internet and probative value of evidence in court. This issue is concerned by Maras and Alexandrou [5]. Because of its impact on social media, deepfake video that has been invested by organizations such as Facebook and Microsoft [6].

As mentioned above, the purpose of the thesis is to study the overview of the existing methods that create fake videos by using Image Processing and Machine Learning technique. In other words, the observational research will be approached. We research the process and technological aspects of deepfake videos and then collect dataset. That helps us to have overview of operational procedure and process in generating video forgeries. As a result, we are going to research a method that can detect the manipulated videos, which implies that we will use experimental research. We formulate the hypothesis to expose the deepfake then test this hypothesis by doing experiments with dataset. In the preliminary research, the video resolution is 1920x1080, frames of video have face portrait and includes only one face.

1.3 Objects and Scopes of research

1.3.1 Objects

Objects of this thesis is to understand the overview of technique and algorithm that are used to produce fake videos. After that, the realistic quality of fake videos will be observed to develop the algorithm. Because we need large dataset to test our algorithm, available dataset has been already published. To solve the problem, we propose two hypotheses, the first is that if there are any artifacts left on image after swapping the face. The second hypothesis is that if there is the difference of video behaviour of real video and fake video. The solution for the first hypothesis is examining the pixel where the face is swapped. And the method for second hypothesis is that we will examine the behaviour of the video over time by consider the difference between frames. We will mention about the methodologies of detecting deepfake for videos in dataset on Internet clearly in chapter 0. In addition, we found the Face Forensics dataset [7] that contains over one thousand forged videos and the original, the other dataset is from the HOHA dataset [8] that contains deepfake videos. After that, we will observe our result to assess the effectiveness of the proposed method and compare the strong point and also weak point with another methods. We will estimate the performance of our algorithm.

1.3.2 Scopes

This thesis includes some insights in the field Image and Video Processing, Machine Learning. We apply Image Processing knowledge that includes algorithm to process videos and images. In addition, knowledge about hardware device such as GPU and CPU and many software is necessary to understand the deepfake technology such as CUDA Toolkit from NVIDIA [9], OpenCL [10], Keras [11], TensorFlow [12]. Nonetheless, the research only concentrates on detection algorithm. As mentioned above, deepfake videos can include the manipulated videos by human or ML algorithms. The videos handled by human is selective editing, which refers to the real videos, however, the content is cut and pasted by many real videos. And ML algorithm can generate videos in which the people say the word that they did not say in real video, for instance, avatar animation is a sample of this kind of video. The fake video of USA President Obama [13] illustrating avatar animation is published. This video will not be detected by proposed algorithms because the face of actor is not swapped with anyone else. In addition, in the dataset, there are many videos that do not satisfy the condition, the camera should direct to the face, we have to filter in aforementioned datasets to extract the swapping face videos to test. Besides, it is possible to realize fake video by the consistency of sound and frame. However, we will not cover the sound analysis and processing in this study.

1.4 Structure of thesis

In this section, the structure of thesis is discussed. In the first chapter, the introduction to topic is mentioned as above. Chapter 2 discusses about general knowledge of topic and concentrates on existing method to detect fake video. In chapter 0, the implementation of our algorithm that detect the loophole of videos forgeries, the techniques and library we used to build the program will be discussed in detail. Discussion and evaluation are discussed in chapter 4, we are going to clarify the novelty and drawback of our methods. The last chapter is future work.

2 Literature Review

In this chapter, the knowledge that related to our thesis will be reviewed. In the first sections, some definitions and technique of Image Processing are mentioned because our input is image. Video is considered as an array of images. The next section will discuss about the existing methods for detection of image or video forgeries that can be searched in journals and conferences. Our search techniques include backward and forward snowballing. Our selected search keywords are method to detect face swapping images or videos, the searched result gives us the works related to face forensics, swapped face detection using deep learning, using Machine Learning to detect video face swaps, Face swap using Convolutional Neural Network. The results searched from the keyword suggest that we should at least understand what is face swapping image and it would be better to understand the overview technique to create face swap. As mentioned above, we do not dive into face swap technique, the face swapping detection will be focused on. The methods we mainly search are the paper on journals and conferences such as Scopus, arXiv, ResearchGate and the number of cited papers is excluded for these papers.

2.1 Image Processing

The thesis uses insight in Image and Video Processing field that support us to implement the first hypothesis. Therefore, in this section, we mention about basic knowledge of Image Processing such as definition of image, filter, mask, histogram and techniques in processing image to understand further step in detection algorithm. And knowledge of Machine Learning that is related to detection algorithm is also mentioned. After reviewing knowledge, the existing methods of deepfake detection will be mentioned.

2.1.1 Background Knowledge

In simple term, image is a 2D-array that has a number of rows and columns. An element of image is called pixel. To access the value of pixel, two indexes one is row index and the other is column index are need. For example, pixel(i,j) is the pixel of row i and column j. In gray image (black and white image), the value of the pixel is unsigned 8-bit values, hence it has a range from 0 to 255, which represents the level of black or white colour. For colour image, each pixel has 3

values: blue, green, red, each value represents the intensity of colour. OpenCV [14] supports us to create image with different types of pixel value, for instance, integer (CV_8U) and floating point (CV_32F).

In many cases, conversion of color image to grayscale is ubiquitous and helpful, to convert a color from RGB color model to a grayscale representation of its luminance, the gamma compresion function must be removed first via gamma expansion or linearization to transform image to a linear RGB colorspace. Therefore, the appropriate weighted sum can be used to the linear color components RGB to estimate linear luminance. Three particular coefficients corresponding to be used for three color, which represent the intensity perception of typical trichromat humans to light of the precise additive primary colors. The coefficient value for green is 0.7152 because human vision is most sensitive to green, the least sensitive to blue, its coefficient value is 0.0722. The coefficient value for red is 1 – 0.7152 – 0.0722.

Filter [15] is used frequently in Image Processing; its usage is to reduce the amplitude of the image variations. To achieve this goal, there is one simple method that replace each pixel by the average value of the pixels around. This kind of filter is also called box filter. The kernel or mask of 3x3 box filter [15]:

Table 1. Box filter 3 X 3.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

In some cases, it is helpful to emphasize the importance of closer pixels in the neighbourhood of a pixel. Therefore, the nearby pixels are assigned a larger weight than the further when computing a weighted average. This can be achieved by using Gaussian function or bell-shaped function.

Histogram [15] is a table that present the number of pixels that have a given value in an image. Therefore, histogram of a gray image will have 256 entries (or bins). Bin 0 gives the number of pixels that have value 0, bin 1 will have value 1 pixel and until bin 255. Histogram can also be normalized such that sum of bins is 1. Simply, histogram gives the distribution of pixel values across the image, which constitutes an important characteristic of the image. Therefore, histogram can be used to characterize content of image and detect specific objects or textures of an image. In addition, the contrast of an image can be improved by stretching its histogram, this make image occupy the full range of available intensity values. The equality of using all available pixel intensities is considered as a good quality of image because in many cases, the visual deficiency of an image is affected by using some intensity values more frequently than others. Therefore, this idea stands behind the concept of histogram equalization that makes histogram of image as flat as possible. Figure 1 shows the histogram of image before equalization, and histogram of image after equalization is represented as in Figure 2.

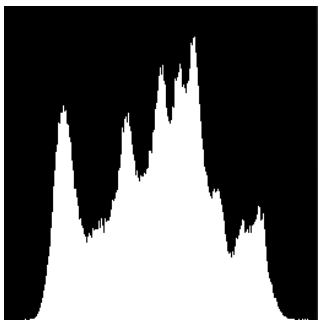

Figure 1. Histogram of image before equalization.

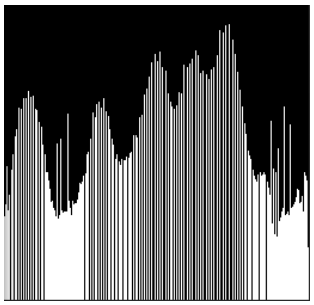

Figure 2. Histogram of image after equalization.

Segmentation is a technique in Image Processing, used to divide colour of image into regions. There are many researches in segmentation algorithm but we are going to apply Efficient Graph-

Based Image Segmentation [16] in our detection algorithm. This method uses minimum spanning tree (MST) is mentioned. In general, MST is used to segment the input image by the difference between a pixel and its adjacent. This algorithm uses L2 norm or Euclidean distance between each colour channels of the pixels as the weight. The algorithm helps us to segment the face region that have difference in colour into group, which support us to detect the face inconsistency region. Therefore, we can detect the position where the face is swapped.

Detecting line plays an instrumental part in our detection algorithm. We apply Hough Transformation [17] for this task. The key point of Hough Transformation (HT) is that the line is the set of pixels on it. As can be seen in Figure 3 [18], line A is determined by the set of the points (3 red points).

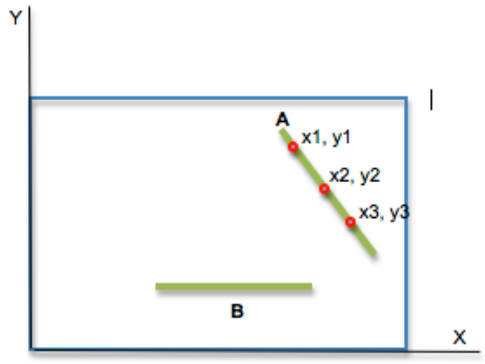

Figure 3. Illustration of line.

The HT represents line in polar coordinate system as shown in Figure 4 [18]. That means the point (x,y) on line will be transformed to be (r,θ), with r which is the length of line (blue) perpendicular to the line we need to find is the distance from center and θ is the angle between blue line and X axis.

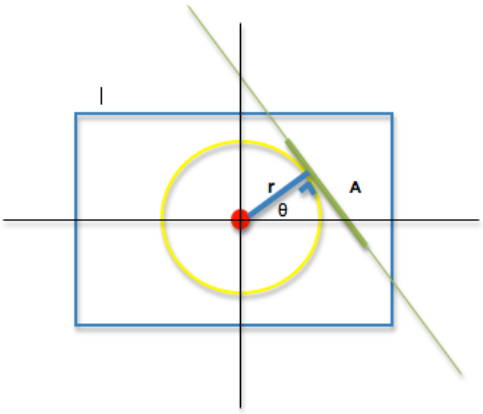

Figure 4. Polar coordinate system.

Therefore, every line can be represented by a unique r and θ and every point on a line will be transformed to exact same r and θ for this line. Furthermore, if the θ is from 1 to 180 and r can be negative then we can transform any lines. We can imagine in Figure 4, if r is negative, we will get line A_2 which is in opposite side of A and parallel to A. We need an element to record the polar coordinate of line, it is called Accumulator which is the 2 – dimension matrix with number of columns is maximum of θ and row is maximum of r.

The reason we use the line detection is that when we do filter edge experiment, we found nearly straight line that represent for unnatural edge on face. Base on this artifact, we can divulge if the video is fake or not. The idea of Hough Transformation is very interesting, the method represents line in polar coordinate system that includes an angle and a radius, one line is presented by one specific angle and radius of circle that has center is the center of image.

2.1.2 Machine Learning

As mentioned above, ML is used to detect swapped video. The first question is that what is ML. The purpose of ML is to convert data into helpful information. For example, assume that we know this video contents swapped face, can we extract some hints from this video to identify if other video contents swapped face or not? Next, some concepts of ML [19] will be discussed. The basic concept is features that are the significant information such as on face image, edge of faces is important. Supervised learning [20] is the method that an algorithm is used to learn the mapping function from the input to output. The purpose is to estimate the mapping function so that when the input is new, the function can predict the output for the input. It is named

supervised learning because the correct answer is known, the algorithm predicts the output for training data and is supervised by the teacher. Supervised learning can be assembled into regression and classification problems. When the output is a category such as “fake video” or “real video”, it is called classification. If the output is a real value such as “90% real” or “80% real”, it is called regression. For our case, supervised learning can be used for the image that we know the face in image is swapped.

Unsupervised learning is the method that has input but there is no corresponding output. The purpose is to model the fundamental structure or distribution of input data to learn more about the data. It is named unsupervised learning because there is no teacher and no correct answer. Unsupervised learning can be assembled into clustering and association. Clustering is used to group data by a given criteria. Association is used to discover the rules that extend the range of data. For our case, unsupervised learning can be used for the image that we do not know the face in image is swapped or not.

The next concept is advanced techniques, it is about Generative Adversarial Networks (GANs) because this technique will be mentioned. However, we are not going to explore it greatly, we just need to understand the purpose of GANs. Basically, the idea of GAN [21] is that there are two agents with the opposing purposes. To be more specific, imagine that the first agent is criminal and the other is investigator. The objective of criminal is to counterfeit the face image so realistically as much as possible that the police cannot differentiate between fake face image and real face image. In the contrary, the police come up with method to differentiate between fake face image and real face image. This process is called an Adversarial Process. GANs makes use of Adversarial Processes to train two Neural Networks that compete to each other until reaching a satisfactory equilibrium. The first network is Generator Network that make effort to generate new face image closed to the given face image dataset. The second is Discriminator Network that try to distinguish between new generated face image and real face image.

2.1.3 Face Identification

Because we will examine the face on image to check if there are any artifacts that tell whether the image contents the swapped face or not. Then, face recognition techniques are used to locate

the position of face on image. The face recognition that uses Histogram of Oriented Gradients (HOGs) [22] is utilized to extract faces from source images. In this research, the authors state that the method is simple but powerful for face detection. HOGs which are image descriptors invariant to rotating 2D image is used in many applications. Deniz et al. use this algorithm for their case, at first, the image is divided into small connected regions that are call cells. For each cell, a histogram of edge orientations is calculated. If the gradient is unsigned, the histogram channels are spread over from 0 to 180 degree and from 0 to 360 degree for signed gradient. The histogram counts are normalized to compensate for illumination. Then, extracting descriptors from salient points is used to achieve invariance to scale and rotation. They find the dominant gradient orientation; all the orientation counts are made relative to this dominant direction. Deniz et al. propose to normalize the face at first step then extract HOG features from a regular grid. Then, they move on to capture important structure to recognize face by fusion of HOG descriptors. Finally, they investigate an approach to make robust use of HOG features. In simple term, this algorithm is illustrated for us to be understandable from the original tutorial [23].

Figure 5. Face sample.

The colour image Figure 5 [23] is converted to gray image Figure 6 [23] and then each pixel is examined. The neighbours surrounding of the pixel are used to find the direction that present the way of image become darker. The direction is represented by an arrow and point from light to dark as in Figure 7.

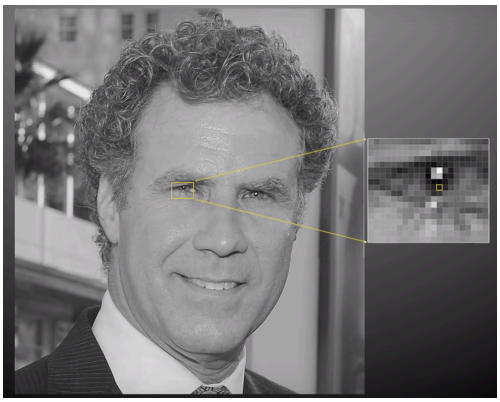

Figure 6. Examination of pixel.

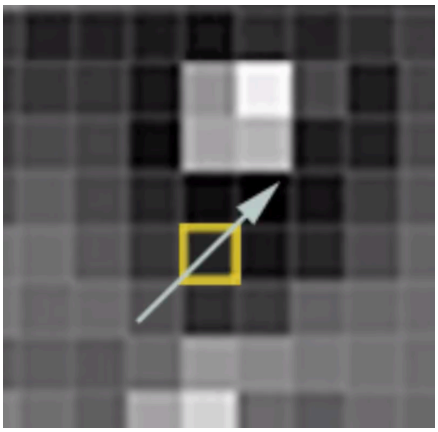

Figure 7. Detecting the direction.

The algorithm only considers the direction of brightness change and divide the image into smaller squares. In each square, the number of directions is counted and the most counted direction will be replaced for this square.

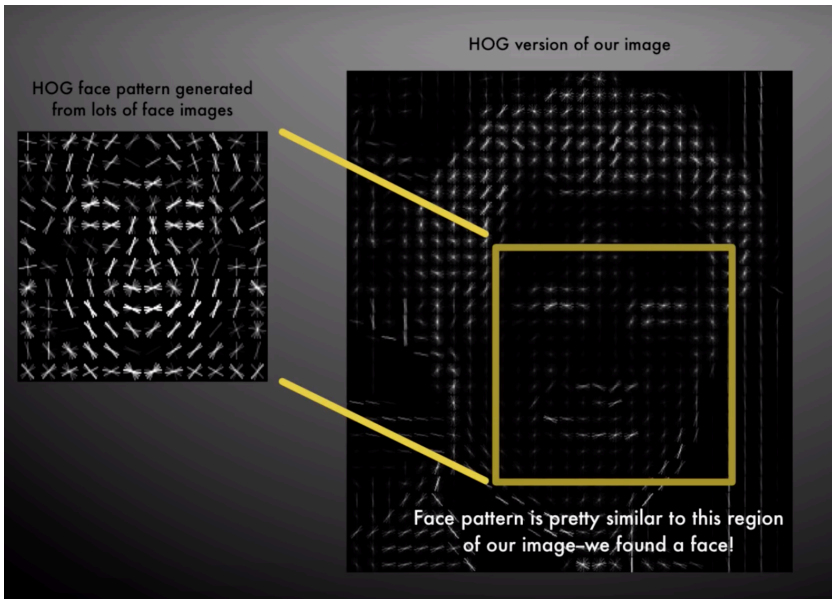

Figure 8. Face structure detection.

The algorithm can expose the very simple structure of the face as in Figure 8 [23].

2.2 Deepfake Generation

Some researchers apply Machine Learning in Image Processing to swap the face of a person to the face of another person. This technique [24] that we found by keyword swapping face source code is implemented by autoencoder, especially, autoencoder has been applied for low dimensional representation of image. The face of the first actor is cropped and aligned to his or her face. After that, autoencoder learns how to encode and decode the face. Decoding face is reconstructing the face. The main goal is that the error of face reconstruction is minimized. The similar action will be done with the face of second actor, the autoencoder. The important part is that the same encoder is used for both actor 1 and actor 2. The reason we need to understand this technique is that we attempt to detect the meticulous pixel where the face is swapped. Therefore, we need to identify the size of the face because we do not know the size of face is swapped. This technique reveals the common image size used to swap face, which is valuable for us to detect swapped face.

The schema of encoder and decode is described in Figure 9 [23] and Figure 10 [23].

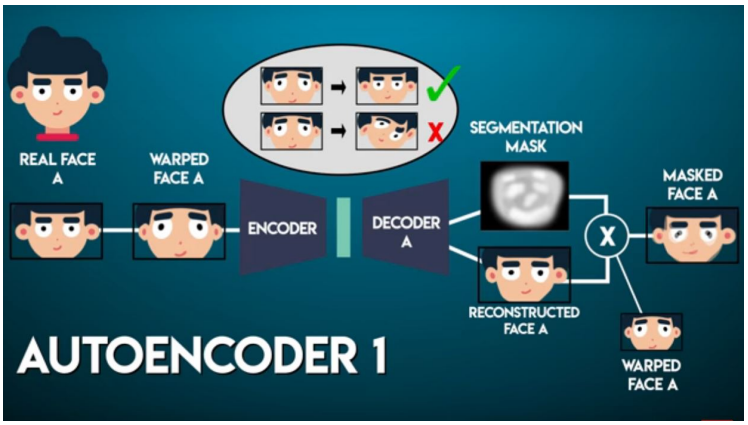

Figure 9. Schema of encoder and decoder.

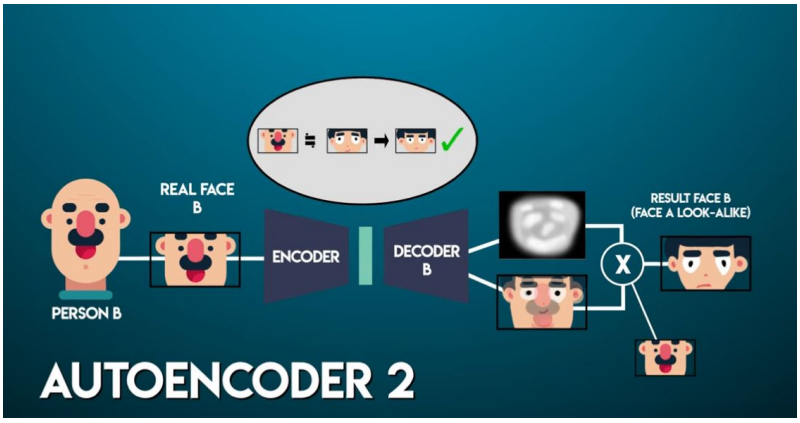

Figure 10. Schema of second encoder and decoder.

Auto encoder purposes to find a function $f(x) \approx x$, where x is the image (the size in autoencoder is selected to be 64 by 64 with 3 channels (RGB), that means the number of variables is $64 \times 64 \times 3 = 12\,288$). Auto encoder can be considered as compressor that parse the image and then figure the data more useful than others, called pattern. After that, the compressor encode image on less bit. The auto encoder goals to encode image in a smaller representation (the size is selected to be 8 by 8 with 512 channels). After completing encoding, the decoder is used to get back the original image from compressed representation. The compression and decompression should not lose any information. However, autoencoder is lossy compression because the image after decompression is approximate to the original image. The representation in mathematic formula of encoder and decoder is that

$$f(x) = decoder\big(encoder(x)\big) \approx x$$

The left-hand side x is an image and usually convolution operations might be need to use. The right-hand side x shows the same face with different angle, lightening condition and many other conditions. It implies that the face needs to decompose into its atomic components, such as the shape of the nose, smile, wrinkle.

2.3 Refinements and Anti-forensics

In recent years, AI-based video synthesis algorithms that are based on deep learning models is developing, especially the Generative Adversarial Networks [25]. Some researchers create deepfake by using GAN, such as Shen et al. [26]. In their work, "Deep Fakes is a popular image synthesis technique based on artificial intelligence". This technique can generate images without given paired training data, which make more power than traditional image-to-image translation. The purpose of deepfake is to capture common characteristics from a set of existed images and to figure out a way of enduing other images with those characteristics such as shapes and styles. They implement GAN for object transfiguration and style transfer. The techniques are leverage that creates difficulty in detecting deepfake.

For the past decade, the neural network has become drawn attention by the researcher because of its power in every walk of life. Some researchers apply CNN that uses neural network as a part of face swapped detection, we will mention about their method in the next section. To understand idea of CNN, we are going to mention about neural network because this is precedent essential of CNN. The knowledge is from original tutorial of Victor Zhou [27]. At first, neurons are the basic unit of a neural network and is described as in Figure 11. A neuron gets input, applies mathematic and create output.

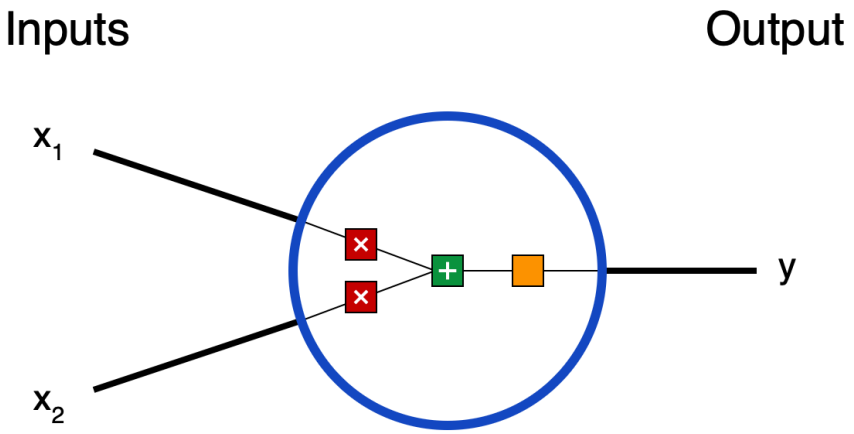

Figure 11. Illustration of Neuron.

The blue circle will process the input. The inputs will be multiplied by a weight, which is presented by the red square.

$$x_1 \rightarrow x_1 \times w_1$$

$$x_2 \rightarrow x_2 \times w_2$$

Next step is that the bias that is presented by green square will be added.

$$x_1 \times w_1 + x_2 \times w_2 + b$$

Then activation function, is symbolized as orange square, is used for the previous result. The activation function purpose to convert an unbounded input value into predictable form output. Sigmoid function is commonly used for activation function. Sigmoid function compresses negative value to 0 and positive value to 1.

$$y = f(x_1 \times w_1 + x_2 \times w_2 + b)$$

At this point, the neuron is built. Next, neurons will be combined into neural network. The simple neural network is like the Figure 12.

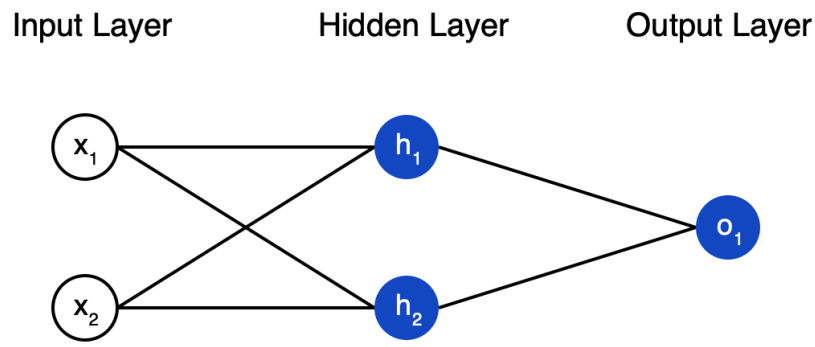

Figure 12. Simple neural network.

The simple network includes 2 input, 2 neurons that contain h_1 and h_2 in hidden layer and 1 neuron o_1 in an output layer. This is called network because outputs from h_1 and h_2 are inputs for o_1. Let's do an example for this network. Assume all neurons use same weights $w = [0,1], b = 0$, same sigmoid activation function $f(x) = \frac{1}{1+e^{-x}}$ and $x = [2,3]$

$$h_1 = h_2 = f(w \times x + b) = f\big((0 \times 2) + (1 \times 3) + 0\big) = f(3) = 0.9526$$

$$o_1 = f(w \times [h_1, h_2] + b) = f\big((0 \times h_1) + (1 \times h_2) + 0\big) = f(0.9526) = 0.7216$$

A neural network can have many numbers of layers that include many numbers of neurons in those layers. Up to this point, the basic of neural network is introduced. Next, the concept of Convolutional Neural Network (CNN) [27] will be presented. To understand CNN, let's examine the classical problem: how machine decide whether in the image of pet, there is a dog or cat. The reason of using CNN because image size is large. If the image size is 224 x 224, colour then the number of features is 224 x 224 x 3 = 150528. If there are 1000 nodes in hidden layer, the network will train over 150 million weights. CNN is neural networks that use Convolutional layers. To understand CNN, a concrete example that is about classification of handwritten digit is considered. Given an image like Figure 13, how the machine can classify digit and given a digit image that is out of dataset, how can machine decide what the digit is. Using CNN is one of solution for these problems. Convolution is one of many techniques in Image Processing. Figure 14 shows the operation of convolution in detail. The source image is processed by a filter or mask to generate the destination image.

Figure 13. Sample images of MNIST dataset.

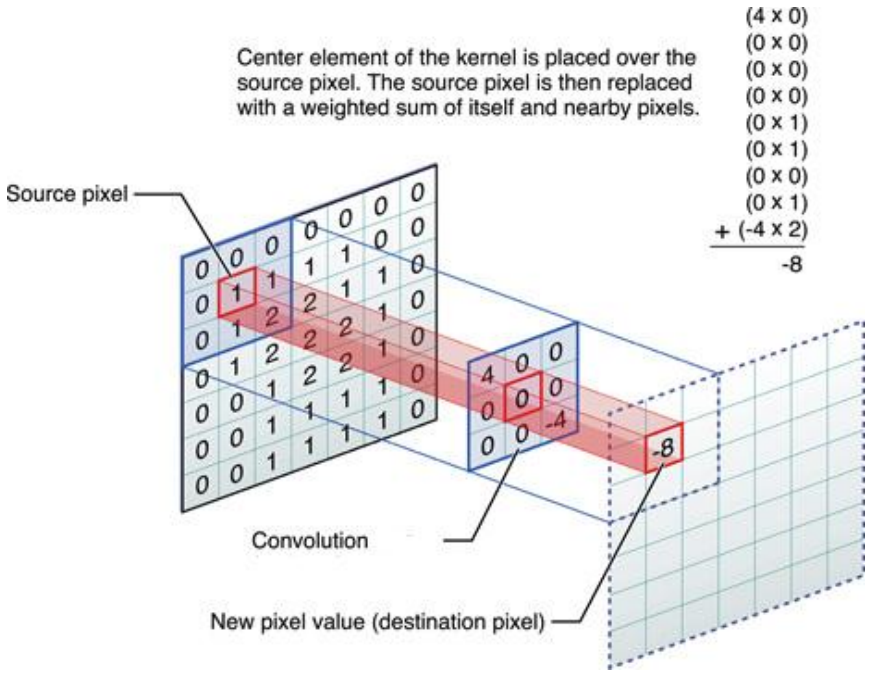

Figure 14. Illustration of convolution.

The filter used in processing the dataset is Sobel filter [19].

-1	0	1
-2	0	2
-1	0	1

Figure 15. Vertical Sobel filter.

Figure 16. Convolved image with vertical Sobel filter.

1	2	1
0	0	0
-1	-2	-1

Figure 17. Horizontal Sobel filter.

Figure 18. Convolved image with horizontal Sobel filter.

Figure 15 shows the vertical Sobel mask size 3x3 and Figure 16 shows the result image of convolution with vertical Sobel mask. The corresponding mask and result of convolution with horizontal direction are displayed in Figure 17, Figure 18. The reason the Sobel filters are used in this situation is that these filters can detect the edge of object. The next concept is Pooling, the reason of using pooling is that in an image, neighbouring pixels might have similar values, then convolution layer might generate similar values for neighbouring pixels for outputs. That means the information in convolution layer might be redundant. The method is reducing the size of the input by pooling value. Figure 19 shows the operation of Max Pooling with the pool size of 2, the matrix on the left-hand side is the source image with size 4x4, the matrix on the right-hand

side is the output with size 2x2. The size of image is 4x4 that is divided by pooling size that is 2, turn the source image into 4 smaller images, then the convolution will be applied for each smaller image. The result is shown as in Figure 20.

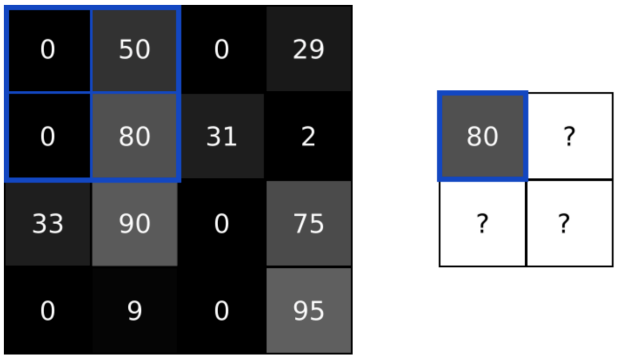

Figure 19. Max Pooling with pool size 2.

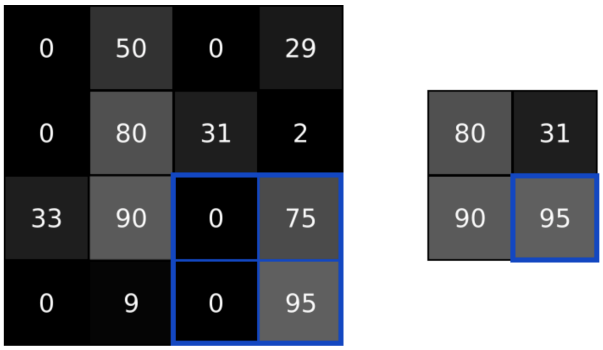

Figure 20. Result applying Max Pool.

The CNN should be able to make prediction. The Softmax layer [27] can applied for this purpose, this layer use Softmax function that are used to turn arbitrary value into probability as activation function. The function is

$$s(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

The meaning of the function is that the network returns a probability value. For example, in this problem, the given digit 4. The network should return 90% of confidence the digit is 4 and 10% the digit is 9.

To sum up, CNN model is created as in Figure 21. The source image has size 28x28, the convolution layer has 8 filters, pooling size is 2 and 10 nodes that represent for 10 digits (from 0 to 9 as in dataset) in Softmax layer.

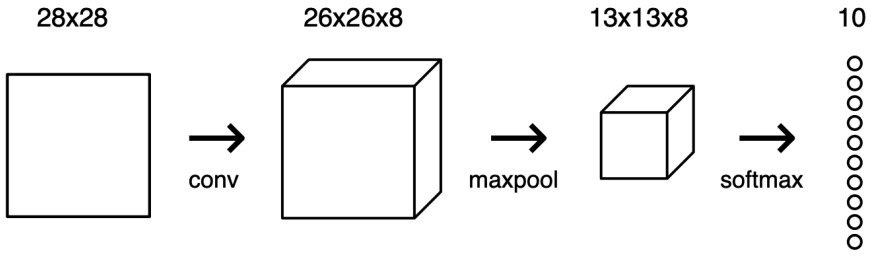

Figure 21. CNN model.

In conclusion, the main idea of swapping face that includes face detection and the basic idea of CNN are pointed out.

2.4 Existing Techniques for Deepfake Recognition

Over the past decades, many researchers have developed many information forensic techniques to identify the authenticity of digital images [28]. The key findings of our research include detecting deepfake images or videos. The realism of state-of-the-art image manipulation technologies are examined and the paper [29] shows the difficulty to detect deepfake image. Many researchers apply Machine Learning to address this problem; However, the main problem of utilizing Machine Learning technique is the processing time and complexity algorithm. Some approaches method of using deep learning to detect face tampering in video automatically and effectively. Their method concentrates on expose the forgeries that used two techniques: deepfake and Face2Face [30]. Afchar et al. [31] utilize mesoscopic level of analysis to detect forged videos. In their opinion, microscopic analysis that bases on image noise cannot be used because image noise is degraded intensely in a compressed video context. Afchar et al. produce network that begins with four layers of successive convolution and pooling in sequence and a dense network with one hidden layer follow the network. The method operates to detect successfully 98% for deepfake and 95% for Face2Face. Amerini et al. [32] apply optical flow fields to exploit inter-frame differences. The hypothesis of the author is that the discrepancies in

motion over frames can be exploited by optical flow that is a vector field computed on two consecutive frames. The optical flow matrices might introduce fake or unusual movements of eyes, lips or even the whole face. After that, the researchers CNN classifiers to learn these features. The result of preliminary research that processed on FaceForensics + + dataset [29] performs promisingly.

Some researchers take advantage of human intelligence to discover altered image. Shetinger et al. [33] conduct an experiment, in this experiment, many users are asked if image is authentic or not after observation and provide evidence to support the answer. In this research, 17208 answers are collected from 393 volunteers and using 177 images from public forensic databases. The results indicate that human vision is not decent in differentiating original from edited images. Li et al. [34] describe a new method to expose falsified face in videos generated with deep neural network. The method is based on the physiological signal that detects eye blinking because synthesized fake videos do not present this signal well. Yang et al. [35] use the inconsistency in head pose to detect fake videos. McCloskey et al. [36] analyse the structure of GAN and show that the treatment in colour of network is markedly different from camera image in two ways. The method is trying to expose artifact to detect GAN generated image like mismatched eye colour. Furthermore, the method demonstrates effective discrimination between GAN generated image and real camera images. Li et al. [37] propose method that detect face warping artifacts to expose deepfake videos. The idea of method is that only limited resolutions image can be created by present deepfake algorithm. After that, the images need warping to match the source face in the original video. Therefore, some unique artifacts will be left by such transforms in the generated videos. The authors of the paper show that usage of convolutional neural networks (CNN) can capture those artifacts effectively. The researchers also show the advantages of the method is that with simple image processing techniques, the artifacts can be simulated. The techniques purpose to turn images into negative example that demands the resource and time to generate by training deepfake model. The researchers point out that the proposed method is more robust to others and is effective in practice. The severe impact of deepfake is mentioned again in the paper of Gu et al. [38] and forensic technique is introduced. The hypothesis of technique is that facial expression and movements of one speaker can be distinctive. Thus, the method tracks

facial and head movements and in next step, the presence and strength of specific action units are extracted. After that, a detection model is built to distinguish one person from another person.

3 Methodology

In this chapter, the research methods and the implementation of our method will be discussed in detail. In the first section, we will mention about data collection and analysis methods. After that, the appropriate research method will be selected for the thesis in our perspective. The detail of implementation will be shown in the second section.

3.1 Methods

At first, to approach the research methods appropriately, we need to collect and analyse data vigilantly. Then we observe the video to check if we can expose hints that inculpate the video. Therefore, we use observational research as the first stage in this research. After that, the hypothesis that describes the artifact to recognize deepfake videos is proposed and the hypothesis need to be tested with the video dataset to verify the confidentiality of hypothesis. In other words, we utilize the experimental research for the second stage. Our methodology intends to investigate artifacts which can be used to identify face swapped videos. Our hypothesis is that the techniques use frame-by-frame spatial mapping for each frame, there may be anomalies from one frame to the next which can be exploited to identify video modification. To be more specific, when swapping the faces, the face of another image is replaced with the face of original image. The fake image can generate artifacts that are able to be found at the contour of the face. We name it unnatural edge. Therefore, we target to find the unnatural edge on face by calculating the first derivative of the pixel at the contour to observe the differences and test the first hypothesis.

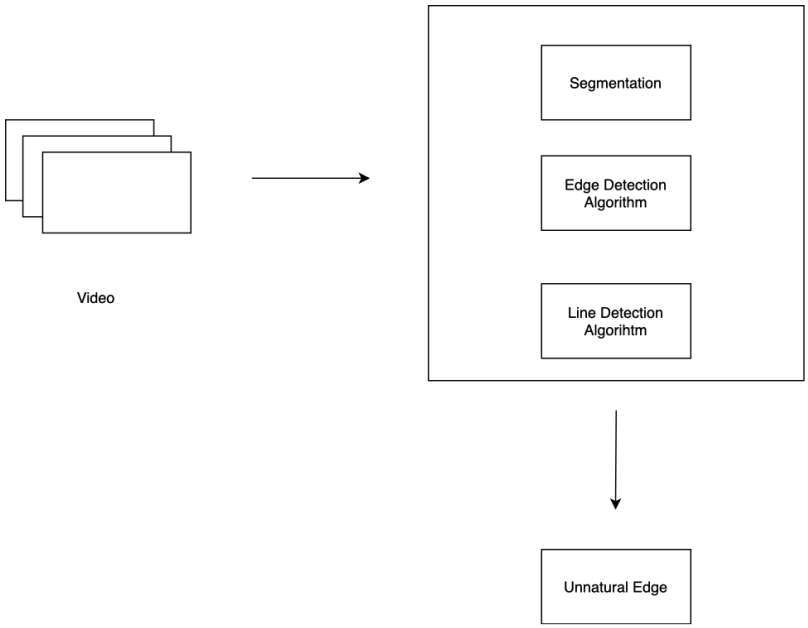

Figure 22. The model of detecting unnatural edge method.

The model of the detection algorithm is described as in Figure 22. From the video input, we extract frame of video and then use combination of segmentation and edge detection method to search edge face and unnatural edge point. The number of lines that pass through unnatural edge point are detected, which help us to authenticate the video.

We also propose another method that looks at the behaviour of video over time. That means we consider the level of similarity of frames. The method names Sum of Absolute Differences, shows the sum of absolute differences of all of the frames of a video against all of its other frames. Originally, this method considers that a natural video has a diverse range of frames but in fake video, there might be repetition of some frames. When a same frame is held for a long time, the method can identify the video as potential manipulation. Based on this idea, our hypothesis is that in real video, the face of actors or actresses might change emotional expression suddenly. Therefore, the difference between frames is larger than the face in deepfake that potentially has difficulty in manipulating the sudden change in expressing facial emotion. We will provide the method to test the hypothesis. We are going to discuss the method, assuming that we are considering two frames, in both frames, we extract the red, green, blue value. Then we compute the subtraction value of the red value in second frame with red value in first frame. Similarly, we do this step for green and blue value. Consequently, we have three values for red, green, blue

respectively. After that, we take absolute value of the premeditated value and take sum of three values, then we will get the frame with new value of RGB. As a result, we will observe the value correspond to frame to check if there is any interesting point.

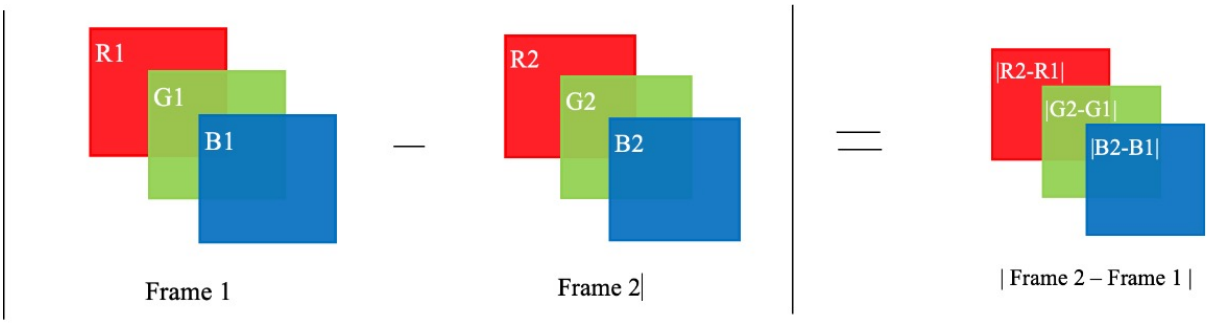

Figure 23. Illustration of second hypothesis.

Secondly, from the result of first step, we can discover some hints to detect the swapping face. It is ready to moving on to next step, in this step, we target to apply Machine Learning to learn how big is difference to identify real or fake video. Then, we would like to use this difference to detect another video.

3.2 Implementation

In this section, we are going to discuss the detection algorithm in detail. As mentioned before, we have two models and the model that analyse behaviour of video over time will be explained as in Table 2.

Table 2. Algorithm: analysing behaviour of video over time.

Algorithm: analysing behaviour of video over time

```
1. For each frame i of video
2.     For every next frame j of frame i
3.         Calculate SAD(i,j)
4.         SAD(i,j) = SAD(j,i)
5.         Store value in matrix
6.         Convert to colormap
```

The function calculate SAD is mentioned in the methods section. As in the previous example, we consider two frames, similarly, the algorithm will examine the frame i and j (i, j is the index of frame), frame i and j are considered as the first and second frame in the example respectively. Then, we are going to save value of SAD into a matrix that represents the difference between frames in a number. Because we use matrix to visualize the result, we implement line 4 to fill all value into matrix, otherwise we just have a half of matrix. In line 6, we purpose to convert number to colour, which makes us straightforward to obtain the large change. The result of the method is interesting. As in Figure 24, there is a large difference between frame 143 and the other frame in the original video. The line connecting the upper left with lower right is diagonal of matrix, the color is symmetric through the diagonal.

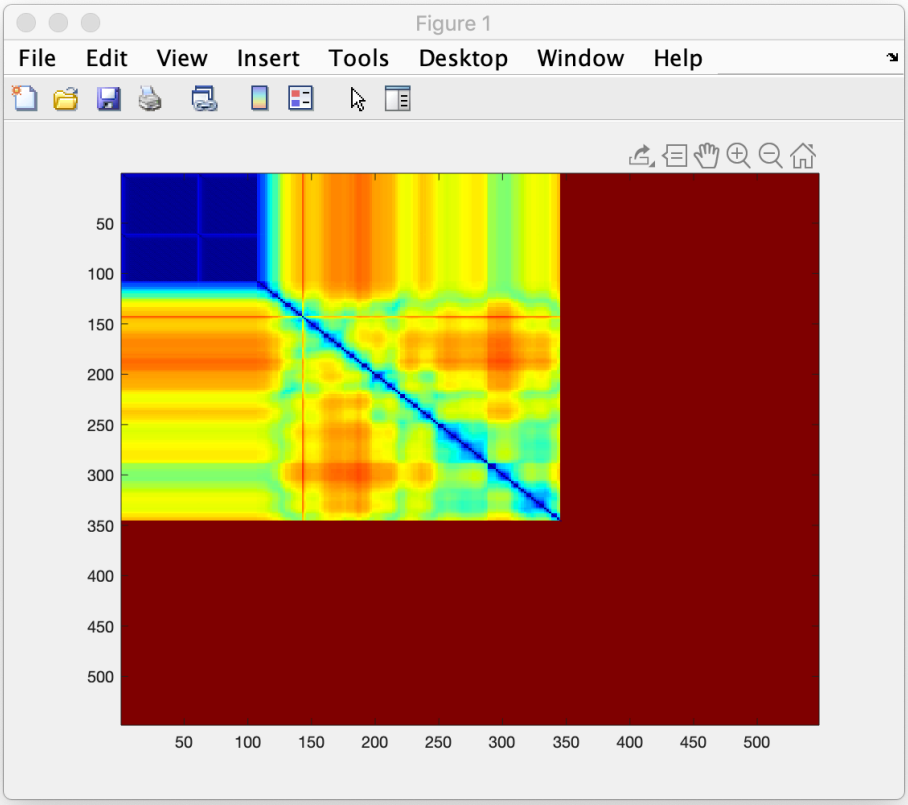

Figure 24. Result of SAD in original video 1.

Figure 25 shows the result of the fake video that is manipulated by deepfake technology in original video 1. We name original video 1 because the video is recorded from Youtube. The

name fake video 1 means the content is from the original but the face of actor in the video is swapped with another person. As can be seen, there is no significant difference between frames, from one frame to next frame, the change is slight and gradual. Therefore, in our preliminary result, we can extract the feature that differentiates fake video from the originals. However, to check this feature, we are going to do more experiment for other videos.

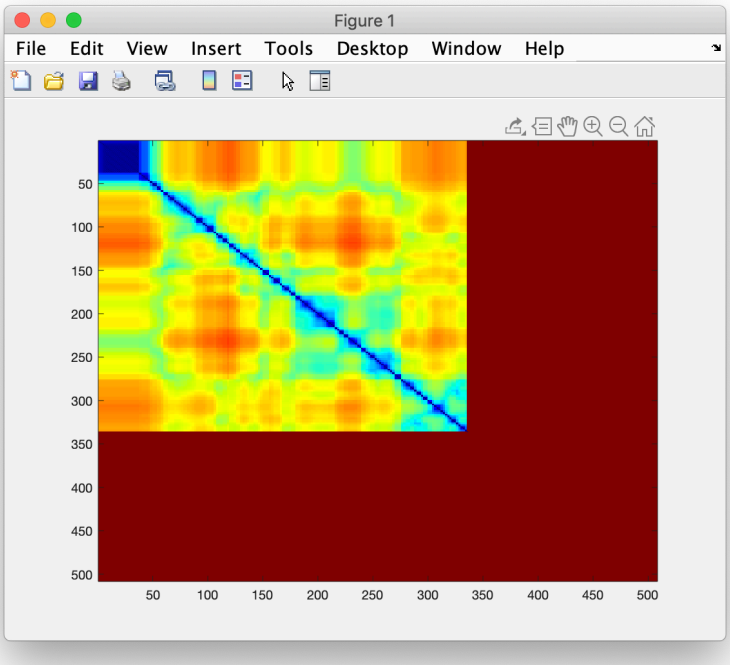

Figure 25. Result of SAD for fake video 1.

The diagonal of the matrix is blue line because the difference between one frame and itself is 0, when we visualize the difference in colormap, blue represents for 0. We discover that the difference is from the change in emotional expression of actors or actress in video. In original video, they usually change facial expression that create remarkable change in frame, which is hardly seen in deepfake video. However, this method is not correct in every case of deepfake videos because in some other original videos, the large difference is not detected as in Figure 26. In addition, to identify how big is difference that can tell the video fake or real is a challenge. We have utilized the neuron network that is described in chapter 2, we calculate the average change and search the biggest difference to predict the confidentiality of videos because our idea is that

the distance between average and largest difference is valuable. Therefore, we use the method to test three videos that include two original videos and one deepfake video. In human vision, it is straightforward to identify if there are large variations as in Figure 24. However, teaching computer learn how to identify the change in colour is an obstacle. Because of this reason, we are going to develop the unnatural edge detection algorithm, the detail is discussed in Table 8.

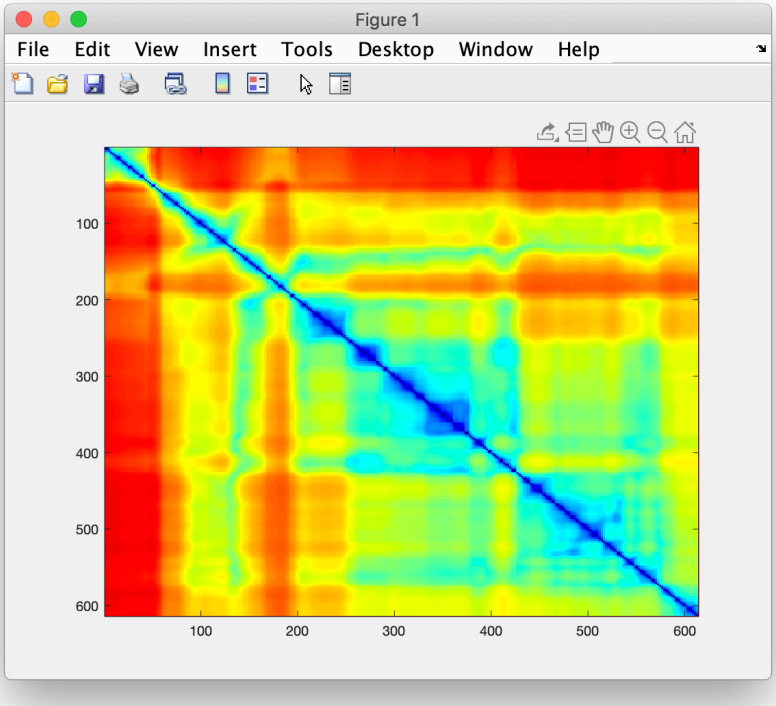

Figure 26. Result of SAD for original video 2.

To implement the unnatural edge detection algorithm, we need to detect the face of actor or actress in video, we utilize the existing face detection algorithm that provide us result not as good as our expectation, our expectation is that the boundary of image fix with the size of face. Figure 27 shows the result, as can be seen, the background appears in the image, which can affect to the result because the detected line can come from background and we can see the unnatural edge on face with human vision. At present, this can be considered as the aperture of deepfake video. Therefore, we take advantage of this point to expose deepfake video. Our purpose is to find the edge face that is the intersection of face and ear and the unnatural edge point is closed to edge face. Then we will check if there is line passing through the unnatural

edge point. The number of unnatural edges will be counted to identify if the video is fake or not because it is always existed on face, this number is expected to be high.

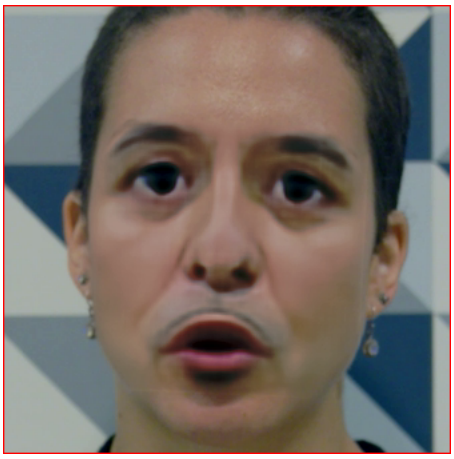

Figure 27. Result of face detection algorithm.

To detect the edge face, we use 15 first frames of video, we use segmentation algorithm to find edge face. The result is show in Figure 28, we make use of the difference in colour between pixels at edge face is high and the position is in the last quarter of image to identify the position of edge face. Therefore, we are going to detect the first pixel having difference value in colour that is greater than threshold and in the last quarter of image, this pixel is considered as edge face. This is the first step and also important progress in our algorithm because if we cannot detect edge face correctly, the unnatural edge point will not be uncovered. That can make our method failed in detecting deepfake video.

Figure 28. Sample result of segmentation.

Our detection algorithm is going to expose the unnatural edge that is represented in Figure 30. As can be seen in Figure 30, there is a vertical line near the edge of face, this line refers to unnatural edge but it is quite blur to see it clearly. Thus, unnatural edge is stressed with red line to make it visible as in Figure 31. Therefore, we are going to detect this line. The image is obtained by using edge filter algorithm, we utilize the convolution method that is mentioned in chapter 2 with a mask that is shown in Table 5. The reason we use this mask is that edge is considered as the pixels where the change in colour is detected. For instance, to detect change for vertical edge, the backward difference and forward difference are used to calculate the sum of difference that is represented as in Figure 29.

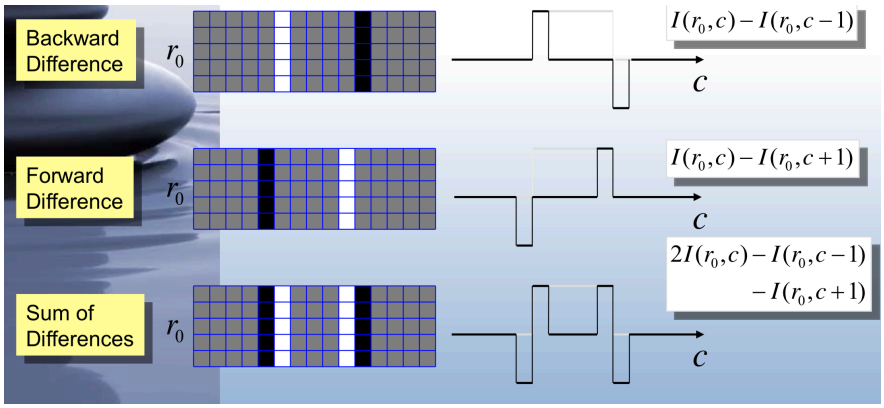

Figure 29. Sum of difference.

Therefore, for vertical edge, the mask that is described as in Table 3 is used.

Table 3. Vertical edge filter mask.

-1	2	-1

Similarly, for horizontal edge, the mask that is described as in Table 4 is used.

Table 4. Horizontal edge filter mask.

-1	2	-1

To sum up, we detect vertical, horizontal and two diagonal edge, that leads us to the edge filter mask shown in Table 5.

Table 5. Edge filter mask.

-1	-1	-1
-1	8	-1
-1	-1	-1

Figure 30. Example of unnatural edge.

Figure 31. Unnatural edge is stressed with red line.

Table 6. Edge Face Detection.

Algorithm: edge face detection

```
1. For each first 15 frames of video
2. ┌    Segmentation
3. │    Detect edge face
   └
```

The segmentation algorithm we used is mentioned in chapter 2.

Table 7. Detect Edge Face.

Algorithm: Detect edge face

```
1. For every point from ¾ of image width to end point of image
2. ┌    Calculate the first derivative of pixel (named diff)
3. │    If diff is greater than threshold
4. │    ┌      Found the edge face
5. │    │      Break function
   │    │
   └
```

For edge face detection as in Table 7, we are going to examine the range of pixels in the last quarter of image because we have observed the result of face detection, the face is relatively in the middle of image. The deviation happens when actors or actresses move quickly, which exerts serve impact on detecting edge face algorithm. When the person moves head rapidly, the edge face is out of last quarter of image. Therefore, the detected point that supposes to be edge face is not edge face point actually, this point can be edge of ear or on background.

Table 8. Unnatural edge detection algorithm.

Algorithm: Unnatural edge detection

```
1. Edge face detection
2. For each frame of video
3. ⌈    Edge filter
4. |    Segmentation
5. |    Detect unnatural edge point
6. ⌊    Detect line that pass through unnatural edge point
```

Table 9. Detect unnatural edge point.

Algorithm: Detect unnatural edge point

```
1. Detect edge face
2. Calculate first derivative of pixels in edge filter image (name diff)
3. If diff is greater than threshold and the pixel is closed to edge face
   point
4. ⌈    Push position of pixel to array
   ⌊
```

This algorithm is implemented with support of OpenCV4.2 [14]. The techniques are used to implement the method are described as in Table 10.

Table 10. OpenCV techniques support for implementation.

Function	Purpose	Library
Canny	Finds edges in an image using the Canny algorithm	imgproc.hpp
Line	The function line draws the line segment between point 1 and point 2 in the image.	imgproc.hpp
blur	Blurs an image using the normalized box filter	imgproc.hpp
GaussianBlur	Blurs an image using a Gaussian filter	imgproc.hpp
imread	Loads an image from a file	imgcodecs.hpp
getCPUTickCount	Calculate the processing time	utility.hpp
VideoCapture	Capture all frames in video	videoio.hpp
cvtColor	Convert image from colour space to another colour space	imgproc.hpp

We have discussed the algorithm in detail. Next, we are going to test it with the dataset. As mentioned before, there are some videos that do not meet the requirement because the camera needs to direct into the face and there is only one face in video. Because of this reason, we need to choose the videos that satisfy these conditions.

Table 11. Testing result of unnatural edge detection.

Video name	Duration	Processing time	Number of detected unnatural edge/detected edge face	Video type
01_11__talking_against_wall__9229VVZ3.mp4	35s	563s	86/128	fake
07__talking_against_wall.mp4	35s	539s	3/398	original
13__talking_against_wall.mp4	44s	719s	122/406	original
04_06__kitchen_still__ZK95PQDE.mp4	35s	448s	9/54	fake

We have run the algorithm with some original and deepfake videos, the result is shown as in Table 11. For instance, the first video in the table is deepfake and the method reveals 86 unnatural edge and 128 edge face. As mentioned above, the method attempts to detect edge face first, then search unnatural edge that is closed to edge face. As our expectation, the frequency of

unnatural edge is large in deepfake video and this number should be low in original video as in the second video. However, in the third video, the number is 122/406, which is not as good as our expectation. By observation of result, it is reasonable for us to state that if the ratio of the detected unnatural edge/detected edge face is over 0.5, we classify the video into deepfake video. Otherwise, the video will be original video. The disadvantage of the method is enlightened when we test the fourth video, the video is fake but the ratio is less than 0.5. This causes us to classify the video incorrectly. And the reason is also related to the drawback that we have tested with the third video. We will clarify the shortcoming of our method in chapter 4.

4 Discussion and Evaluation

In this chapter, we are going to discuss about our method. In the second section, the novelty in this research will be justified, the main contribution and drawback of the study will be mentioned.

4.1 Discussion

A common argument with deepfake is that the underlying AI that generates it is evolving. If we try to detect deepfake with further AI, we will end up with an escalating adversarial situation where the GAN identifies flaws which the generator the repairs, and so on. This means that deepfake evolves rapidly because two machines are arguing with each other.

However, there is a problem with this argument because the adversarial conflict is only as good as its programming, and we are asking the reasonable question of whether there are artifacts which survive. In particular, we are looking at artifacts within a frame image, such as unnatural line edges; there are some artifacts that are strange as behaviour over time (from the perspective of a human reviewer), such as body movement not matching facial expression; and there are some artifacts that can be measured over time but which aren't necessarily apparent to a human viewer, such as glitches in mapping faces from one frame to the next.

We go into this research realising that we have not got good results, in part because deepfake is getting better all the time. The main objective is to better understand what we can identify to provide more knowledge about how we might develop detection tools. In particular, we are using image and video processing rather than further machine learning, so that we can understand and describe the processes we are using, rather than relying on an underlying machine learning algorithm running on trained data which we can't necessarily analyse.

The unnatural edge detection is implemented in C++ and OpenCV using Xcode11. We run the test on MacOS with 2.7 GHz Intel Core i5 processor, 8 GB 1867 MHz DDR3 memory, Intel Iris Graphics 6100 1536 MB graphics. The techniques for the algorithm are from Image Processing

field, as discussed in chapter 0, the results are shown there. In addition, we have plan to apply ML to solve the shortcoming of our method. Our preliminary idea is that we would like to define concept of unnatural edge and reinforce the ML model to learn the concept.

The analysis of video behaviour is implemented in Matlab with the same machine.

4.2 Evaluation

4.2.1 Novelty

The first hypothesis builds up the first novelty in our method, which is we utilize the artifacts left on swapped face image. In the previous works, Li et al. [37] expect to reveal artifacts in affine face warping. In our perspective, the artifacts we target to is similar to this paper but we concentrate on examining the disparity of pixels where the contour of face is swapped. Our algorithm concentrates on solving the processing time of video and the limited resources. The detection methods that use ML spend much time labelling data and training data, therefore, to implement the ML detection, large dataset and huge processing time are required. Our proposed method is looking for the feature that highlights limitation of deepfake. In other words, the method depends on the artifact we name unnatural edge that presents in video, which helps us overcome the ML obstacles that are about dataset and processing time. We have mentioned the time processing of 5 sample videos in chapter 0.

The analysis of video behaviour is understandable and simple to implement. This method can be used for low resolution quality videos and the method will be effective if the actors or actresses change emotional expression on face suddenly in video. However, the idea has been originally used to detect frame or scene repetition, we tend to specialize the algorithm in detecting deepfake video.

4.2.2 Drawback

As every garden has its weed, our proposed method has some disadvantages that we have already revealed partly. In this section, we are going to summarize all problems of the algorithm. The

first algorithm is unnatural edge detection, this method means for high resolution videos. As mentioned in chapter 0, we base on first 15 frames of video to identify the edge face point. In these frames, if face oscillates with large amplitude, the edge face point is likely failed to be detected, which causes the failure for algorithm. The algorithm takes advantage of combination of segmentation, edge filter, derivative calculation, line detection. All these has a weak point in common, that is using the constant value. The constant value in segmentation is to define how big is the difference in colour to divide pixels into a group. In edge filter and derivative calculation, we use threshold to decide if the pixels are edge face point or not, this is similar to unnatural edge. If the threshold is not appropriate, edge face and unnatural edge cannot be detected. Line detection also needs threshold value to identify how large is the set of point to become a line. And the fixed value has serious problem that is not able to adapt to every data. In other words, at present, our algorithm using fixed threshold value cannot work perfectly for every video in dataset. The problem-solving direction is promising because at present, deepfake technique is not excellent enough to conceal the unnatural edge. However, it is very hard to find a specific value that can be used for all videos that have variables in colour, camera direction, face shape, face colour. That explains the reason why the result in chapter 0 is not as good as our expectation. In addition, background in face detection is not removed completely, therefore the line detection can make mistake that is face does not have unnatural edge but the line detected is from background. In addition, if the unnatural edge is blotchy, the line can’t be detected.

The second algorithm is the analysis of video behaviour. We have revealed the disadvantages of this method before, we aggregate all in this section. As discussed before, this method base on the difference between frames to detect if video is fake or real, the large difference can be recognized easily by human vision after we use colormap to visualize the difference in term of color. However, to define how large is the number to classify the video is an obstacle. This method will be failed to detect real video if the person does not change emotional expression suddenly. These videos happen in real life such as political interview or serious conference.

5 Future work

In chapter 4, we have discussed advantages and disadvantages of two proposed method. In this chapter, we will mention the direction that means to overcome the drawback of these methods. For unnatural edge detection, the suggestion is that we would like to specialize the face detection algorithm for deepfake detection because the background appears on image. At present, we reuse the face detection algorithm that means for general purpose; therefore, we would like to make it adaptive to our target. The big aperture is using the fixed threshold value, which is the biggest challenge for us. We are going to do more experiments to search a dynamic formula to generate threshold value that can be adaptive to many videos. Until now, we cannot give an exact number to show the effectiveness of our method because of these limitations. We can also improve the algorithm by identifying the exact pixels on face and after that, line detection is progressed on those pixels, which saves processing time and make result more accurate.

About the analysis of video behaviour, this seems a promising method but we need to search more assessments to improve the effectiveness. We have tried to find the relative distance between maximum value and average value to check if these values are helpful to reveal any clues about the video but it seems not optimistic. Therefore, we have to supply more clues to solve the problem.

6 Summary

In the thesis, we research about deepfake that has severe impact on information security. In particular, we concentrate on detecting face swapped videos. We have studied the knowledge of Image Processing and Machine Learning that support us to understand the overview of deepfake forensic and anti-forensic in chapter 2. Based on the knowledge and dataset observation, we have implemented two methods to detect face swapped videos. The requirement of dataset is discussed clearly in section 1.3. We present the detail of algorithms in pseudo code that support us to visualize program. After implementing the algorithms, we have evaluated the algorithms to clarify the advantages and disadvantages. Then, the future work is suggested to improve the thesis.

References

[1] R. Margaret, “Whatls.com,” September 2015. [Online]. Available: https://whatis.techtarget.com/definition/multiprogramming.

[2] FaceApp, “https://www.faceapp.com,” [Online]. Available: https://www.faceapp.com. [Accessed 3 November 2019].

[3] mrdeepfakes, “https://github.com/iperov/DeepFaceLab,” [Online]. Available: https://github.com/iperov/DeepFaceLab. [Accessed 3 November 2019].

[4] Fortunelords, “36 Mind Blowing YouTube Facts, Figures and Statistics – 2017 (re-post),” 13 December 2017. [Online]. Available: http://videonitch.com/2017/12/13/36-mind-blowing-youtube-facts-figures-statistics-2017-re-post/. [Accessed 27 October 2019].

[5] M. Marie-Helen and A. Alex, “Determining authenticity of video evidence in the age of artificial intelligence and in the wake of Deepfake videos,” *The International Journal of Evidence & Proof,* vol. 23, no. 3, pp. 255-262, 2019.

[6] C. Elizabeth, “Facebook, Microsoft launch contest to detect deepfake videos,” 6 September 2019. [Online]. Available: https://www.reuters.com/article/us-facebook-microsoft-deepfakes/facebook-microsoft-launch-contest-to-detect-deepfake-videos-idUSKCN1VQ2T5. [Accessed 27 October 2019].

[7] A. Rössler, C. Davide, V. Luisa, R. Christian, T. Justus and N. Matthias, “Faceforensics: A large-scale video dataset for forgery detection in human faces,” *ArXiv,* vol. abs/1803.09179, 2018.

[8] L. Ivan, M. Marcin, S. Cordelia and R. Benjamin, “Learning realistic human actions from movies,” in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, 2008.

[9] NVIDIA Corporation, “https://developer.nvidia.com/cuda-zone,” NVIDIA Corporation, 2019. [Online]. Available: https://developer.nvidia.com/cuda-zone. [Accessed 7 November 2019].

[10] NVIDIA Corporation, “https://developer.nvidia.com/opencl,” NVIDIA Corporation, 2019. [Online]. Available: https://developer.nvidia.com/opencl. [Accessed 7 November 2019].

[11] keras-team, “https://keras.io,” keras-team, [Online]. Available: https://keras.io. [Accessed 7 November 2019].

[12] A. Martín, A. Ashish, B. Paul, B. Eugene, C. Zhifeng, C. Craig, S. C. Greg, D. Andy, D. Jeffrey and D. e. a. Matthieu, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” *ArXiv,* vol. abs/1603.04467, 2015.

[13] BuzzFeedVideo, “You Won’t Believe What Obama Says In This Video,” 17 April 2018. [Online]. Available: https://www.youtube.com/watch?v=cQ54GDm1eL0. [Accessed 17 April 2020].

[14] OpenCV team, “OpenCV 4.2.0,” 23 December 2019. [Online]. Available:

https://opencv.org/opencv-4-2-0/. [Accessed 8 April 2020].

[15] L. Robert, OpenCV 2 Computer Vision Application Programming Cookbook, Birmingham: Packt Publishing Ltd.

[16] P. Felzenszwalb and D. Huttenlocher, "Efficient Graph-Based Image Segmentation," *International Journal of Computer Vision,* vol. 59, p. 167–181, 2004.

[17] D. Richard and H. Peter, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM,* vol. 15, 1972.

[18] K. Bruno, "Hough Transformation - C++ Implementation," 3 May 2013. [Online]. Available: http://www.keymolen.com/2013/05/hough-transformation-c-implementation.html. [Accessed 8 April 2020].

[19] K. Adrian and B. Gary, Learning OpenCV: Computer Vision with the OpenCV Library, 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc., 2008.

[20] B. Jason, "Supervised and Unsupervised Machine Learning Algorithms," 16 March 2016. [Online]. Available: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/. [Accessed 8 December 2019].

[21] H. Aadil, "Building a simple Generative Adversarial Network (GAN) using TensorFlow," 8 May 2018. [Online]. Available: https://blog.paperspace.com/implementing-gans-in-tensorflow/. [Accessed 9 December 2019].

[22] D. Oscar, B. Gloria, S. Jesus and D. L. T. Fernando, "Face recognition using Histograms of Oriented Gradients," *Pattern Recognition Letters,* vol. 32, no. 12, pp. 1598-1603, 20 January 2011.

[23] R. Siraj, "DeepFakes Explained," 2 February 2018. [Online]. Available: https://www.youtube.com/watch?v=7XchCsYtYMQ&feature=youtu.be. [Accessed 7 December 2019].

[24] V. Olivier, "https://github.com/OValery16/swap-face," 15 February 2018. [Online]. Available: https://github.com/OValery16/swap-face. [Accessed 2 November 2019].

[25] G. Ian, P.-A. Jean, M. Mehdi, X. Bing, W.-F. David, O. Sherjil, C. Aaron and B. Yoshua, "Generative Adversarial Nets," in *Neural Information Processing Systems 27 (NIPS 2014)*, 2014.

[26] S. Tianxiang, L. Ruixian, B. Ju and L. Zheng, "" Deep Fakes " using Generative Adversarial Networks (GAN)," Tianxiang, Shen, San Diego, 2018.

[27] Z. Victor, "Machine Learning for Beginners: An Introduction to Neural Networks," 24 July 2019. [Online]. Available: https://victorzhou.com/blog/intro-to-neural-networks/. [Accessed 7 December 2019].

[28] C. S. Matthew, W. Min and J. R. L. K., "Information Forensics: An Overview of the First Decade," *IEEE Access,* vol. 1, pp. 167-200, 2013.

[29] R. Andreas, C. Davide, V. Luisa, R. Christian, T. Justus and N. Matthias, "FaceForensics++: Learning to Detect Manipulated Facial Images," *ArXiv 2019,* vol. abs/1901.08971, 2019.

[30] T. Justus, Z. Michael, S. Marc, T. Christian and N. Matthias, “Face2Face: Real-Time Face Capture and Reenactment of RGB Videos,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 2016.

[31] A. Darius, N. Vincent, Y. Junichi and I. Echizen, “MesoNet: a Compact Facial Video Forgery Detection Network,” in *hal-01867298, version 1*, 2018.

[32] A. Irene, G. Leonardo, C. Roberto and D. B. Alberto, “Deepfake Video Detection through Optical Flow Based CNN,” in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[33] S. Victor, M. O. Manuel, d. S. Roberto and T. J. Carvalho, “Humans are easily fooled by digital images,” *Computers & Graphics,* vol. 68, pp. 142-151, 2017.

[34] L. Yuezun, C. Ming-Ching and L. Siwei, “In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking,” in *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, Hong Kong, 2018.

[35] Y. Xin, L. Yuezun and S. Lyu, “Exposing Deep Fakes Using Inconsistent Head Poses,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, 2019.

[36] M. Scott and A. Michael, “Detecting GAN-generated Imagery using Color Cues,” *ArXiv 2018,* vol. abs/1812.08247, 2018.

[37] L. Yuezun and L. Siwei, “Exposing DeepFake Videos By Detecting Face Warping Artifacts,” *Yuezun, Li; Siwei, Lyu,* vol. arXiv:1811.00656, 2019.

[38] A. Shruti, F. Hany, G. Yuming, H. Mingming, N. Koki and L. Hao, “Protecting World Leaders Against Deep Fakes,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019.