

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
T Ü R K İ Y E



SIGNATURE RECOGNITION USING MACHINE LEARNING

Shalaw Mshir ABDALLAH

Master's Thesis

DEPARTMENT OF SOFTWARE ENGINEERING

AUGUST 2020

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
T Ü R K İ Y E

Department of Software Engineering

Master's Thesis

SIGNATURE RECOGNITION USING MACHINE LEARNING

Author

Shalaw Mshir ABDALLAH

Supervisor

Assoc. Dr.Mehmet KAYA

AUGUST 2020

ELAZIG

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
T Ü R K İ Y E

Department of Software Engineering

Master's Thesis

Title: Signature Recognition Using Machine Learning

Author: Shalaw Mshir ABDALLAH

Submission Date: 22 July 2020

Defense Date: 13 August 2020

THESIS APPROVAL

This thesis, which was prepared according to the thesis writing rules of the Graduate School of Natural and Applied Sciences, Fırat University, was evaluated by the committee members who have signed the following signatures and was unanimously approved after the defense exam made open to the academic audience.

Supervisor:	Assoc. Dr.Mehmet KAYA Syracuse University	<i>Signature</i> Approved
Chair:	Prof. Dr. Asaf VAROL Firat University, Technology Faculty	Approved
Member:	Assoc. Dr. Muhammed Fatih TALU İnönü University, Faculty of Engineering	Approved

This thesis was approved by the Administrative Board of the Graduate School on

..... / / 20

Signature

Assoc. Dr. Kürşat Esat ALYAMAÇ
Director of the Graduate School

DECLARATION

I hereby declare that I wrote this Master's Thesis titled "Signature Recognition Using Machine Learning" in consistent with the thesis writing guide of the Graduate School of Natural and Applied Sciences, Firat University. I also declare that all information in it is correct, that I acted according to scientific ethics in producing and presenting the findings, cited all the references I used, express all institutions or organizations or persons who supported the thesis financially. I have never used the data and information I provide here in order to get a degree in any way.

13 August 2020

Shalaw Mshir ABDALLAH



PREFACE

This research stems primarily from my love for producing more reliable ways of understanding biometrics techniques. Biometrics is described as the art of interactively and statistically measuring and studying physiological data, in particular, that experienced from human lives, so we are working with human lives. I am seriously grateful to several noble persons, without them, I couldn't have had the chance to study out of my country. From those peoples I learn too much since this four-course, without them I would not have reached this stage. First, my supervisor Assist.Prof. Mehmet KAYA whose helps me too much and I take too advantage from his ideas. Then to our best teacher Prof. Asaf VAROL, whom I learn too much from his skills and his experience. And our new Chair of the Department prof. Dr. Resul daş which helps us too much. My classmates for their helping and for these beautiful days in a beautiful city Elazig.

Shalaw Mshir ABDALLAH
ELAZIG, 2020

TABLE OF CONTENTS

	Page
PREFACE	IV
TABLE OF CONTENTS	V
ABSTRACT	VII
ÖZET	VIII
LIST OF FIGURES	IX
LIST OF TABLES	X
ABBREVIATIONS	XI
1. INTRODUCTION	1
1.1. The History of Informatio Security.....	2
1.2 Machine Learning and Categories.....	2
1.2.1. Supervised Learning.	3
1.2.2. Unsupervised Learning	4
2. AN OVERVIEW OF SIGNATURE RECOGNITION	5
2.1 Nature of Human Signature	6
2.2. Explanation	7
2.3. Signature recognition Technique - Offline and Online	7
2.3.1. Offline Signature Recognition Method.....	8
2.3.2. Online Signature Recognition Method	8
2.4. Types of signature forgery	9
2.5. Handwritten Signature Verification (HSV) Applications	10
2.6. Siamese Network & Triplet Loss.....	11
3. RELATED WORK.....	13
4. MATERIAL AND METHOD	16
4.1. Dataset	16
4.2. Pre-processing	16
4.3. Convolutional Siamese Network (CSN)	17
4.3.1. Convolutional Layer	18
4.3.2. Pooling Layer.....	21
4.3.3. Triplet Loss Classification	22
4.4. Ubuntu Operating System.....	23
4.5. Tensor Flow	24
4.6. Opencv.....	25
4.7. Summary.....	25
5. RESULTS AND COMPARISON	26
5.1. Training the Model	26
5.2. Testing the Model	26

5.3. Comparison Table 28

6. CONCLUSIONS AND FUTURE WORK..... 30

6.1. Conclusion 30

6.2. Future Work..... 30

REFERENCES 32

CURRICULUM VITAE



ABSTRACT

Signature Recognition Using Machine Learning

Shalaw Mshir ABDALLAH

Master's Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Software Engineering

August 2020, Page: xi + 34

Signatures are commonly applied as a process of private classification and confirmation; various certifications such as bank checks and legal actions necessitate signature affirmation. Confirming the signatures on a huge number of papers is a complicated and time-consuming duty. As a result, a sensitive extension has been recognized in biometric personal confirmation and authentication systems that interact to unique, quantifiable physical attributes (fingerprints or hand, face, ear, iris, or DNA scans) or observable characteristics (gait, sound, etc.). Some techniques are utilised to represent the proposition system's aptitude to differentiate the real signatures from the copies. This method performs a new procedure for signature verification and recognition, utilising a Kaggle dataset for training the model with a siamese network and triplet loss as a classifier.

Keywords: Signature recognition, Signature verification, Siamese network, Convolutional network, Triplet loss; one-by-one kernel

ÖZET

Makine Öğrenmesini Kullanarak İmza Tanıma

Shalaw Mshir ABDALLAH

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Yazılım Mühendisliği Anabilim Dalı

Ağustos 2020, Sayfa: xi + 34

İmzalar yaygın olarak kişisel tanımlama ve onaylama yöntemi olarak kullanılır. Banka çekleri ve yasal faaliyetler gibi birçok sertifikanın imza doğrulaması gerekir. Çok sayıda kağıdın imzasını doğrulamak karmaşık ve zaman alıcı bir iştir. Sonuç olarak, hassas genişleme, benzersiz ölçülebilir fiziksel özelliklerle ilgili (parmak izleri, el ve yüz, kulak, iris veya DNA taraması) (yürüyüş, ses vb.) biyometrik kişisel doğrulama ve kimlik doğrulama sistemlerinde tanınmıştır. Önerilen sistemin sahtecilikten gelen orijinal imzaları belirleme yeteneğini tanımlamak için çeşitli yöntemler kullanılmaktadır. Bu yaklaşım, modeli bir siyam ağı ve bir sınıflandırıcı olarak Üçlü kaybı ile eğitmek için bir kaggle veri seti kullanarak imza doğrulama ve tanıma için yeni bir teknik sunar.

Anahtar Kelimeler: : İmza tanıma; imza doğrulama; siyam ağı; evrişim ağı; üçlü kayıp, tek tek çekirdek

LIST OF FIGURES

	Page
Figure 2.1. Us biometric technology market size by end 2012-2013	5
Figure 2.2. An instance of genuine signatures and a skilled forgery	8
Figure 2.3. Embedding the signature by CNN processes	11
Figure 4.1. Full siamese network architecture	17
Figure 4.2. Take out a feature map from RGB image.....	18
Figure 4.3. Block image before one by one kernel	20
Figure 4.3. (b) Block image with one by one kernel.....	20
Figure 4.4. Block image displays as a block of embedding	22
Figure 4.5. Final siamese architecture with classification TL.....	23
Figure 5.1. User list interface.....	27
Figure 5.2. Enrollment section to recording the users	27
Figure 5.3. Recognize section to recognizing the signatures	28

LIST OF TABLES

	Page
Table 2.1. An example of types of forgery signature	9
Table 4.1. Commonly used signature datasets	16
Table 4.2. CNN Parameters	19
Table 4.3. Software and Tools used	25
Table 5.1. The result of the Kaggle dataset.....	26
Table 5.2. Comparison of approaches.....	29



ABBREVIATIONS

ANN	: Artificial Neural Network
ML	: Machine Learning
SL	: Supervised Learning
CNN	: Convolutional Neural Network
ReLU	: Rectified linear unit
PL	: Pooling Layer
HSV	: Handwritten Signature Verification
SV	: Signature Verification
OFSR	: Offline Signature Recognition Method
OLSR	: Online Signature Recognition Method
TL	: Triplet loss
FM	: Feature Ma

1. INTRODUCTION

Information security has increased in various study subjects, such as user authentication and authorization, network security, hardware safety, software safety, and data encryption. Information security has grown an essential requirement for preserving virtually all information event purposes. Protection is an essential scientific field with several of its difficulties varied in cooperation with the computer science and engineering areas. Lately, due to the learning range and communication technology, data security has begun to include developing subjects such as cloud computing protection, intelligent city security and separation, health care, the Internet of Things, vehicle internet safety, and different kinds of wireless sensor systems security. Also, information security has increased to spread not only technical safety defects but also communicative and commanding safety tests. Classic operational improvement methods focus on the usability of this method, where protection is given to the end step without preference [1]. However, new plans take the safety process into account as safety is analysed early in the design process. Newly created methods must be completely shielded from possible protection assault. Producing methods like the Internet of Things or health care without enough security can leak sensible data and threaten conditions. Security is the property or condition of safety, freedom from risk, or protection against enemies – those who might aim to abuse intentionally. National security, for instance, is a multi-layered system that keeps the state's independence, estate, sources and people. Obtaining a suitable standard of security for an organisation also demands a multifaceted system. An effective organisation must have the following various security layers to preserve its approaches.

1. Physical security: to preserve physical items, objects or areas from illegal access and abuse.
2. Personnel security: to preserve an individual or group of individuals approved to access the organisation and its progress.
3. Operations security: to preserve the circumstances of a specific process or set of actions.
4. Communications security: to protect communication media, technology and content.
5. Network security: to saves network parts, communications and content.
6. Information security: to preserve the affection, honesty and availability of information estate, whether in storage, processing or transportation, which is achieved through the implementation of policy, education, training, recognition and technology.

The National Security Systems Committee determines data security as the stability of data and its significant components, holding frameworks and devices that utilize, collect and transfer this information [2].

1.1. The History of Information Security

The story of information security starts with computer security. The requirement for computer security – the requisite to secure physical areas, hardware and software from fulminations – occurred during World War II, when the original mainframes, produced to support estimates for delivery code-breaking, were put to use. Various levels of protection were performed to preserve the mainframes and the probity of their data. An approach to fine-tuned army positions, for example, was established using brands, keys and the facial identification of confirmed personnel by protection defenders. The expanding necessity to keep public security ultimately led to more challenging and technologically sophisticated computer security sureties. Throughout these primitive years, information security was an outspoken compute formed predominantly of physical security and manageable certificate distribution systems. The principal threats to security were physical vandalism of equipment, surveillance on the system outcomes, and destruction. One of the first documented security predicaments that happened surface of these classes passed in the early 1960s when a systems manager was operating on a message of the day file and another administrator was editing the key file. A software glitch combined the two files, and the entire key file was written on every output file [3].

1.2. Machine Learning and Categories

Machine learning (ML) allows machines to learn without being explicitly programmed. Another way ML prepares machines to do what people do is learning by practice. ML is a field within the broader discipline of artificial intelligence. These machines continuously learn by separating data to discover patterns so we can expertly recognise malware in encrypted events, detect insider threats, predict where bad blocks are online to keep people secure when browsing or shield data in the cloud by uncovering different user roles.

ML algorithms are classified within four sections: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. Supervised learning involves producing a dataset beside the product that is used to teach the method. Unsupervised learning involves working to discover relationships between the objects in a dataset without holding the correct decisions within the practice; this signifies that the algorithm works to collect details that it believes to have interacted below one label, based only on the analytical features. Semi-supervised learning associates the preceding two models by training the system, employing a dataset holding labelled and unlabelled information objects. The idea is to change design performance by producing the use of both data object kinds. Reinforcement learning, in reverse, utilizes trial and error to define the step collection that metrically maximises any common compensation [4, 5].

1.2.1. Supervised Learning

Supervised learning is a section of ML algorithms that depend on marked practice data. The practice data is a collection of samples; every sample is a set (x, y) where x is an input, and y is the producing value. The algorithm generates a method that could be practised for drafting strange information. Supervised learning produces two main sections: regression and classification algorithms. Classification tries to obtain the most suitable place for the data by setting all input into proper forms so that the product of the prophetic function is discrete. This has the potential benefit of doing one of several types accessible as components of the practice data. The most influential classifications hold the artificial neural network (ANN) and deep learning [6].

ANN is a common supervised analysis algorithm, usually practised when there is enough labelled training data with several characteristics; it requires a non-linear hypothesis purpose. Neural networks hold millions of simulated 'brain cells' that obey and receive in a way amazingly like the human brain. Likewise recognised because of ANNs, neural networks are usually comprised of a software feint that acts like millions of connected mind cells communicating to resolve intricacies, reach judgments and understand models precisely as the human mind can. Usually, one invisible layer poses as an intermediary layer. This layer supports added information from the set of characteristics accessible as portion of the practice data and is called the activation layer.

For instance, banks process credit card gains that may need an automatic way to recognise the signatures of the employees and customers. With reasonable activity inputs and proofs, a neural network can take necessary action to recognise the customer and staff signatures [7].

Deep learning is regarded as a specific instance of supervised ML. Overall, it attempts to create designs discovered in the data utilising a diagram with multiple processing layers. Those treating layers hold assemblies that use linear and non-linear changes in the data to obtain as much valuable data as able. Deep learning is very extremely to ANNs. ANNs can be classified as one of the deep neural networks. Nevertheless, deep learning algorithms are more common as they can be utilised for both labelled and unlabelled data and a large scale of neural networks. Deep learning presently utilises ANNs on an extensive system that can be practised with more data and have more reliable performance. There are several deep learning variations other than ANNs, such as convolutional neural networks (CNNs) and recurrent neural networks [8].

CNNs were founded by LeCun; he transformed image processing and pushed the old-fashioned feature extraction. CNNs work straight on matrices and tensors for images with three RGB colour courses. CNNs are now generally used for image categorisation, image segmentation, object identification and face recognition. One of the various advantages of CNNs is decreasing the number of parameters in ANNs. And the most important theory about problems that are explained by CNNs does not need to focus on where is the feature located. For example, in a face detection

application, we do not need to give notice about where the faces are located in the images [9]. One example of a CNN is Facebook's facial identification. Several years ago, after a photo was uploaded to a profile, Facebook asked users to add a name to the face on the photo manually. Now, Facebook uses CNNs to tag a friend in a photo automatically. Image analysis is the method of applying a picture as input and outputting the image's class, whether it is a cat, dog, bird or a probability of stocks that best represent the image. An image is input within a model; the model detects it as an array of pixel values depending on the image's resolution and size. For a coloured 640×640 JPG image, the characteristic network input array will be 640×640×3, where 3 is the RGB value. Each of the vertical numbers is designated as a value between 0 and 255, representing the pixel depth at that specific point. Among such input, the computer produces numbers that describe the possibility of a picture being in a given class, such as .75 for a dog, .15 for a bird or .10 for a cat.

CNNs possess the following four layers [9].

Convolutional: while the data connects to this layer, the layer convolves each filter over the spatial dimensionality of the input to compose a two-dimensional activation chart.

Pooling: pooling layers attempt to decrease the dimensionality of the image slowly and, therefore, considerably reduce a significant number of parameters and the computational complication of the model.

Fully connected: a fully connected layer includes neurons that are correlated to the neurons in the two neighbouring layers, joined outwardly to any layers inside of them. This is similar to the way that neurons are designed in popular forms of ANN.

It is also recommended to apply rectified linear units within these layers to increase performance.

1.2.2. Unsupervised Learning

Unsupervised learning is another part of ML in which patterns are judged based on unlabelled practice data. The practice data only includes inputs $\times 1, \times 2 \dots \times n$, and unknown outputs. Hence, it tries to understand the practice data by gaining connections and patterns inside it. The main section of unsupervised learning is clustering.

Clustering is the simplest method of understanding a collection of data objects. This presents the data more acceptably as it grants more structure to it by creating a finite collection of groups rather than having an aggregation of random data points. This is very significant in forms such as business division and social system investigation [10].

2. AN OVERVIEW OF SIGNATURE RECOGNITION

Individual verification and classification are actively expanding fields of study and development. Various biometrics techniques are utilised to verify a person's status, which can be expressed in salient (fingerprint, face) and behavioural features (voice, signature). Biometrics is described as the art of interactively and statistically measuring and studying physiological data, in particular, that experienced from human lives. Biometrics is growing more proper and possible as a more confident substitute to popular password security policies; a biometric feature is difficult to steal, duplicate or even guess. The motivation is the rise in the reputation of computerised businesses. For this purpose, a set of biometric applications are being proposed in the field of digital economics and automatic banking policies. Conventional certification methods (e.g. smart cards, passwords and PINs) experience the supervision point of the design. On the other hand, biometric authentication systems are not clearly movable, are different for every person and cannot be misplaced, taken or crushed, as they are fundamental to human lives. The next measures are practised when using an appropriate biometric [11, 12].

1. Uniqueness: how singular is the biometric characteristic?
2. Ease of copying and theft.
3. Acceptability by the audience.
4. Price of implementing the designated biometric data.

The human hand presents many physiological biometric features. Finger and hand geometry are commonly applied. The *Biometrics Market and Industry Report 2013* says that signatures and hand systems are a pair of advanced biometric technologies and held 14% of the business market in 2012–2013 as shown in Figure 2.1.

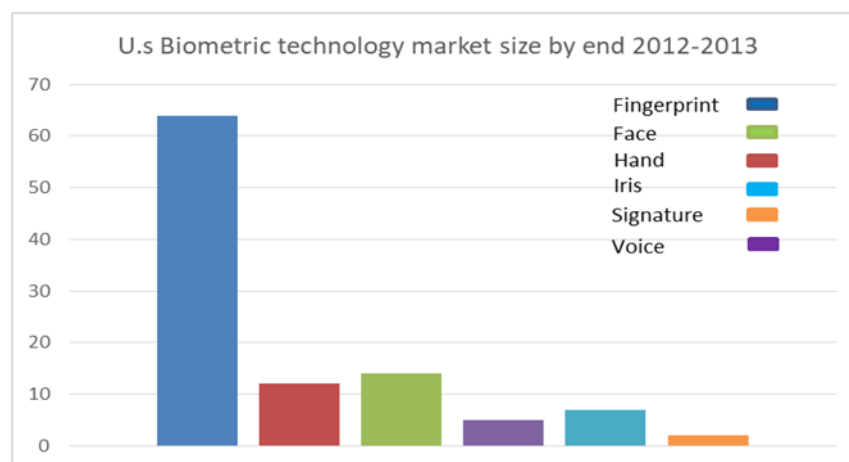


Figure 2.1. US biometric technology market size by the end of 2012–2013 [4]

Today, handwritten signatures are commonly practiced and agreeable manners of biometric certification. There are two instances of signature verification systems: offline and online. Normally, online is more serviceable than offline due to the accessibility of information, such as stroke order, writing speed, or pressure. Nonetheless, this enforcement requires the costs of specific tools for registering the pen tip pathway, bettering the system price and decreasing the original application conditions. In an offline system, only a static picture of the signature is possible; it is further complex than an online system because sufficient information is not accessible, and it is difficult to communicate information's about signatures. Offline signature confirmation rely on recognising patterns; signatures are defined utilising fixed-size characteristic vectors. Those vectors are created using feature extractors and local learning, such as local binary patterns, the histogram of oriented gradients or Gaussian grid features obtained from signature forms. They also use global information like Fourier descriptors, the number of sections in the design, distributions, tortuosity, directions, curvatures and chain codes. There are various methods of signature confirmation; one of them is through applying designs and fundamental pattern differentiation. Graphs support for more important information's that can be helpful for signature confirmation, for instance, utilising edges to take local learning in links and their relationships in the global construction. However, the capacity of graphs comes at the price of huge computational difficulty, which is why graphs have been practiced somewhat infrequently for signature confirmation in the yesterday. In this precept, the model is very diverse; it is an offline signature verification utilising a convolutional siamese network. Siamese networks are dual networks with shared weights, which could be taught to get a feature location where similar marks are established in the vicinity. This is accomplished by exposing the network to a couple of identical and dissimilar measures and degrading the distance within similar sets while improving the far away among dissimilar sets. Analyses carried out on cross-domain datasets show the ability of a network to examine fake in different languages (scripts) and handwriting modes [13].

2.1. Nature of Human Signatures

Pursuant to the American Heritage thesaurus, a signature can be specified as the title of a person signed by his or her hand. Another explanation relates to the complete method of signing, which indicates how the signature is a portion of the signature itself. This points to the assumption that the features of the signing method, for example, pen strength, speed and stroke, are different for each person. The first description defines a signature as a static, two-dimensional picture, which does not include any time-related learning. Another explanation depends on the dynamic trait of the signing method. Handwritten signatures change in several ways, and there are also many differences in the signatures of people in various cultures. For proverb, some people sign their

names while others could utilize only their initials or accept signatures that are merely correlated to their names, and some signatures are challenging while others are easily and quickly forged [14]. Ruth Rostron describes the graphology of a signature utilised to investigate and show the nature of a person. She explains the differences in the sign of various people and more signatures by the identical person, which could change in many parts. Graphology is an art in its own right and is practised in many states in innovative communities [15]. Gubta states that if any signature specialist sees that two signatures by the same person are equal, they could be mechanically recognised as a fake by tracking. From a scientific design subject, signs by the same person may change together globally and locally and could also change in system and adjustment. Regardless of these differences, it is assumed that even nevertheless the strength of the sign somewhat changes, they will still have similar features, such as oblique phases and strength, that distinguish them as original signatures [16]. It has further been proposed that human specialists are highly skilled at recognising forgery but possibly not so in establishing original sign. For example, in a full consider, Herbst and Leo's reviewer noticed signature experts were able to refuse or accept up to 25% of unique signatures with no forgery allowed. New individuals allowed up to 50% of counterfeiting.

2.2. Explanation

The difference between a signature and handwriting is that a signature is “a sign, which that person writes, and is applied to indicate the support of conducting element”, such as a legal agreement, but the handwriting “is an activity or writing method implemented utilising the hand”. The intention of the sign confirmation process is to identify the author of a unique example while the meaning of the signature confirmation method is to approve or refuse a case [17].

2.3. Signature Recognition Techniques – Offline and Online

Signature confirmation practices are classified in the next two sections. Offline systems (also indicated to as static) offer no possible learning at the time of the signature. A scanned picture of the sign is possible.

Online systems (also indicated to as dynamic) include the possibility of time-related data in the form of a p-dimensional purpose of time, where p expresses the numbering of signature features, like pen strength and speed.

Although online systems have been further defined since they maintain signature dynamics as additional information, offline classifications are also necessary for areas where a client is not present at verification time, meaning that there is no possible information explaining the signature

method. For example, signature verification while processing a check payment can only be managed offline because features cannot be extracted online [18].

2.3.1. Offline Signature Recognition Method

Offline signature confirmation achieved the first method utilised to determine the signature confirmation difficulty. It includes discerning real and fake signatures on static models. Different from online process, offline methods only have the static picture holding the sign as an input, outwardly holding any information on the signing method. Some problems that may result from offline methods are associated with the scanning method (yelling on the picture) and the signature purchase method where complex pen points and widths can create several forms. Figure 2.2 represents an instance where the four original signatures on the left are hard to recognise from the fake signature on the right [18].

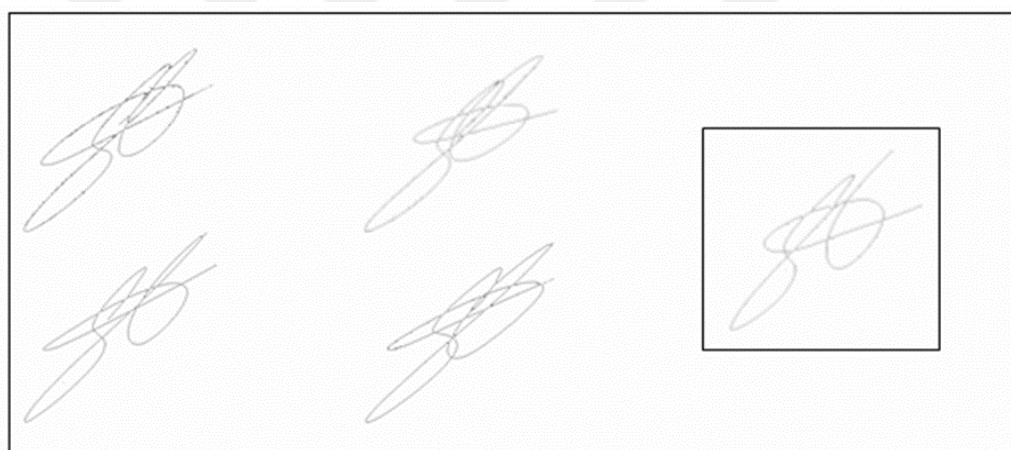


Figure 2.2. An instance of genuine signatures and a skilled forgery [4]

For this purpose, offline systems are perpetually interested in chance and easy fakes, whereas skilled fakes are typically discovered by online techniques. Offline signature confirmation for discovering experienced copies is still an open study problem.

2.3.2. Online Signature Recognition Method

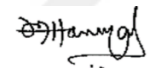
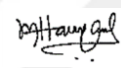
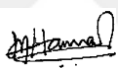

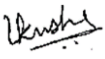







Online signature verification has the dynamic characteristics of the signatures and the truth that online confirmation includes as well as the ability to learn the signature method. Recognition accuracy is more important than the offline process that does not include every description of the signature method information. However, the online system needs specific hardware to estimate the active features of the signing method. A digitising pad is applied for this confidence, which

essentially displays the flight path and velocity of the method typically alongside strength, pen point setting and other features. The mixture of those features is said to be the most different for each person. The basic methodology for both systems is approximately equivalent, typically covering data acquirement, pre-processing, feature extraction, decision-making and performance assessment. Betwixt the two techniques, offline methods appear to be further effective than online methods, but they are further complicated because only static features are accessible from the signature picture [19].

2.3.3. Types of Signature Forgery

The purview of a signature confirmation operation is to identify whichever a presented sign is real or faked. For this purpose, the confirmation processes count on the forgery varieties. The Table 2.1 presents the three main forgery classes.

Table 2.1. An example of types of forgery signature [20]

Genuine	Skilled Forgery	Unskilled Forgery	Random Forgery
			
			
			

Random forgery: the counterfeiter holds no way to the original signature and no knowledge regarding the author's name, so they create an arbitrary signature.

Simple forgery: the faker has no path to the example of the sign except recognising the author's name, and the faker creates the signature in his/her way.

Skilled forgery: the faker has a path to the examples of the original sign, so he/she can copy it.

The three kinds of forgeries discussed before are classified using various verification methods. Usually, random and simple fake classes are recognised utilising offline systems that depend on static characteristics because those systems have confirmed their potential to explain the characteristics compared to signature patterns. Because the offline system needs time-related learning and is not able to model writing movement, it is more difficult to recognise the skilled copies as they have roughly the equivalent form of a specific signature. For this purpose, processes

that depend more on pseudo-dynamic characters are more proper for distinguishing skilled copies, as all can use handwriting motion features [21].

2.4. Handwritten Signature Verification (HSV) Applications

A competent HSV system can be used in various fields of protection purposes. Handwritten signatures are now allowed in communities, and they perform an essential part of business, marketing and judicial proceedings [4].

Financial transactions: a signature is judged as a 'seal of approval' and is the favoured certification instrument because of its suitability. However, the handwritten signature created an attraction for counterfeit warnings and greatly increased financial needs. For example, check forgery has become a significant problem.

Online banking activities: HSV applications are further utilised in Internet banking methods for client confirmation prospects. A digital disk is needed to obtain a user's written sign so the user will not be required to memorise any passwords or PINs. The mixture of static and dynamic characteristics makes the written sign unique for each person; therefore, it is complex to copy.

Check to process: the sign confirmation system can be smoothly combined within the check processing workflow where signs are checked and automatically matched to the individual signs collected in the database. Human interference is only needed if the signature in the problem appears at a particular threshold.

Credit cards: unluckily, because of their dynamic performance, credit cards are too easy to copy. Various methods have been suggested to decrease credit card fraud. Nevertheless, they are furthermore unideal as they have been shown to require competing advantages or safety subjects.

Computer user authentication: HSV methods can replace the common practice of using keys to obtain computer methods, such as administrative systems and data practices. Of course, a standard HSV dynamic system requires connecting a digital tablet to each workstation to obtain signature features.

Passports: HSV systems can be utilised in any possible field. In this sample, a person travels to a licensed office to present an example of the electronically signed signatures on the passport magnetic strip (future passports may have magnetic strips for processing faster). Again, the client is obliged to log in on a whiteboard at the point of entry, and it must be matched to the source signatures collected on the magnetic strip.

2.5. Siamese Network and Triplet Loss

Siamese networks were first launched in the beginning 1990s by Bromley and LeCun to resolve sign confirmation as a trouble with picture matching. The siamese neural network consists of dual networks that accept distinctive inputs but are connected to a power function at the head. This function calculates some metrics during the highest-level characteristic descriptions on every faction. The parameters are linked among the twin networks. A convolutional siamese network is a kind of neural network siamese network in which analogous neural networks are followed by two same-neural networks (CNN). The CNN network itself is a forward-feeding neural network that provides neurons in various dimensions to work on high-dimensional data efficiently. One of the main benefits of utilising CNNs is that they can learn local characteristic indicators and be reasonably powerful enough to distort network inputs. In Figure 2.3, the picture on the left represents the sing of an organisation's staff. Siamese networks accept an input picture and discover the encodings for that picture, then accept the same network outwardly any changes to weights or biases and enter different images without their encodings.

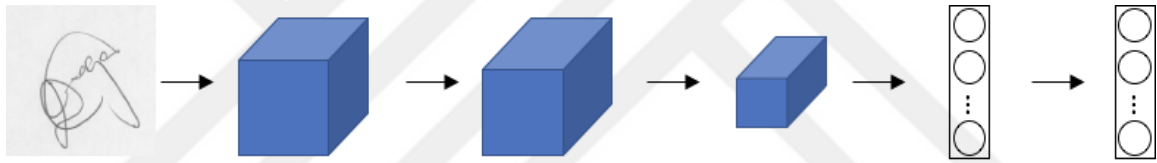


Figure 2.3. Embedding the signature by CNN processes

We match these two encodings to check for similarities among the two pictures. These encodings perform as a distinctive, underlying representation of images. With the related pictures, there are similar features/notations. Using this, we can match and know whether the two pictures have similarities or not.

Some may ask how the network can be trained. A programmer can practice the net by utilizing an anchor image and matching it to both the positive and negative samples. The dissimilarity among the anchor picture and the positive picture must low, and the dissimilarity among the anchor picture and the negative picture it should be high.

$$L = \max (d (a, p) - d (a, n) + \text{margin}, 0) \quad (2.1)$$

The above formula describes the triple loss function. The variable (a) renders the anchor picture, (p) renders the positive picture, and (n) renders the negative picture. The variance among (a) and (p) must be smaller than the variance among (a) and (n). The last variable, named margin, is a hyper factor that is joined to the loss equation. Margin determines how much difference there

should be; that is, if $\text{margin} = 0.2$ and $d(a, p) = 0.5$, then $d(a, n)$ should be at least 0.7 . Margin helps differentiate between the two pictures. Hence, applying this loss function can compute gradients, and with the help of gradients, it can update the siamese weights and biases. To train the network, the net takes an anchor picture and at random chooses positive and negative pictures, calculates the loss function and updates its gradients [22].



3. RELATED WORK

A comprehensive literature study was conducted in the field of signature confirmation process. The Internet was the central research source because many journals and conference procedures can be establish at *IEEE* and *Elsevier Science*. Moreover, *Google Scholar* with the digital literature library *CiteSeer.IST* was the dominant online search engine utilised to search for quotations. Initially, generic keywords were used, as shown below, to get a general feel for the entire region.

1. Signature confirmation
2. Handwritten signature confirmation
3. Handwritten signature confirmation
4. Signature classification

After many exciting results, the general meaning of this area was comprehended. At this point, it was observed that there were two varieties of handwriting confirmation, online and offline, as clarified in Chapter 2. After deciding to concentrate on offline methods, the following words were used to dig deeper into offline signature confirmation techniques:

1. Offline signature confirmation
2. Offline handwritten signature confirmation
3. Offline signature confirmation state of the art
4. Offline signature confirmation rapprochement study

The overhead keyword issues were extensive and interesting. It can be readily learned that this study field dates back several years and as yet represents an unlock study issue. At this point, the study standards also covered the year of publication to discard old articles.

Several surveys on handwriting signature verification systems were attended, and the methodologies were applied. Several methods have been recently introduced, and much of the research has been done for both online and offline signature verification. Following are diverse ways to learn about offline signature verification in the literature.

Sahms [24] introduced a method that runs by segmenting pictures and dividing them within various blocks. The method estimates the relationship of lines within those divisions.

Porwik [25] introduced a different way that depends on three levels of sign verification. The method applies histograms and Hough algorithms to the horizontal and vertical signs. The Hough remodel is applied to explore the path that rises in a sign to recognise the sign's frame. After Hough transformation, an algorithm is applied to decrease the quantity of direct paths that are identical using Hough remodel.

Toscana et al. [23] established an offline signature recognition model utilising five multi-layer perceptions, relying on the characteristics obtained from an offline signature. The model was

examined and judged using 950 signs from 19 people. The outcomes indicated that the introduced method produced a high identification rate, including simplistic signatures.

Macabi et al. [24] provided a system to test written signs, practising the neural network structure. The improved method relied on features obtained from the signatures. The selected features were applied to teach the neural network, and the outcomes revealed that the recommended method produced right outcomes with a total error rate of 33% for the most suitable state.

Pascual-Gaspara et al. [25] introduced a multi-section vector quantisation method for online signature verification. The designers practised the MCYT dataset, which included 330 users and 25 skilled copies per person, operated by five diverse forgers.

J. B. Fasquel and M. Bruynooghe [26] suggested one without an internet connection, signature confirmation method. The system holds three levels. The beginning level is to convert the original sign utilizing personality and four Gabor instructions. The next level is to consult the sign that was dissolved with the likewise modified signs of the training dataset. Later in the third level, validate Signatures by merging the decisions regarding each transfer. The recommended system supported 62.4% of them to be rejected forging practiced for testing when 99% of real signs are identified perfectly.

Sharifa Mumtaz Syed Ahmed and others et al. [27] offer an off-line sign confirmation, developed by mathematical methods. They practiced the hidden Markov Modeling method (HMM) to developing a design for every local characteristic. The confirmation stage consists of three panels of mathematical methods. In the beginning, the HMM-based probability average calculated. Next, this average set in the border ranges approval or refusal through the normalization reception. Then the Bayesian method utilized to determine approval or refusal specific sign example. To blind FAR was 22%, but skillful was 37%.

Shafiei and Rabiee [28] introduced a novel mode of online confirmation signature utilizing the HMM principle that divided each sign based on their perceived POIs and then calculated for each slide several features that are variable in scope and displacement. The resulting sequence then applied to prepare HMM to produce sign confirmation. The method provided results 4% for approval rate and 12% for the refusal rate.

Edson Justino et al., [28] suggested checking the offline signature model using HMM. The characteristics obtained from the segmented picture. Then confirmation applying the HMM structure.

Carr and Fox [29] suggest further than 90 characteristics to consider. When the collection of characteristics is elected, there may be no necessity to save the source sign, and just the characteristic marks for the source sign are required. Furthermore, when an analysis sign is provided, just the characteristics values are required, not the sign. This usually stores on the storage,

which is why the sign representation through a set of its feature marks is sometimes declared as signature compression.

Ding [30] produced a scheme that utilized a closed shape tracking function to describe the edges of the various signs with various closed shapes. The curve data of the closed lines tracked to multi-resolution signals were analyzed utilizing wavelet transformations. Zero crossbreeding to the bending data were extracted as matching features. A mathematical computation was devised to regularly determine the most solid and distinctive closed lines and their connected frequency data. Depend on this data, the maximum threshold mark that checks the accuracy of the characteristic extraction method was estimated. Matching occurred with dynamic torsion of time. Trials were separately managed on two data factions, one for English signs and others for Chinese signs. For specific checks, when skillful frauds are recognized AERs are declared at 13.4% and 9.8% for the associated data sets. When a simply informal copy is recognized, AERs are declared at 2.8% and 3.0%.

Kaewkongka, et al. [31] suggested a technique for identifying an off-line sign practicing a Hough transformation to discover blow lines of the sign pictures. This Transformation was utilized to obtain parameterized space of the distinctive frame as a unique sign feature. In the analysis, the posterior diffusion neural was applied as a mechanism to judge the performance of the suggested technique. The system was tested with 70 test signatures from different people. Experimental outcomes show the identification degree of 95.24%

Xiao and Leedham [32] developed the affirmation of deficient copy practicing the returned neural network classifier to give special consideration to limited fixed portions of signs by weighting the identical node rejoinder by the feedback tool. The effect of the method provided an enhanced tale to classify a skillful copy.

In this paper [33], they represent the task of verifying the signature of a writer by connecting to a Siamese network. These acts accomplished by reducing the Euclidean distance among similar pairs while simultaneously expanding it between different pairs. They train the model among various datasets, for CEDAR the accuracy was 91%, and for GPDS was 88%.

4. MATERIAL AND METHOD

This chapter explains two parts of this study. The first part shows the methodology used to train the system. The second part shows the tools and material to build both the train system and the interface to serving the system. After a lot of research, Tensorflow and OpenCV frameworks decided to build the train section, and the react JavaScript and java spring frameworks choose to implement the interface. Furthermore, this chapter explains the benefits of training the models on the Ubuntu operating system.

4.1. Dataset

The signature acquisition process serves comparable levels for almost all public data sets. The unique signs were gathered in one or more assemblies and needed a user to submit some examples of their signs. The user received a report that included multiple containers and presented a model of his/her sign in every container. The forgery cluster followed the following approaches: users perceived examples of original signatures and were required to replica the signature one or further times. Users who performed copying were not masters in producing copies. After that, the reports were scanned and pre-processed [34, 35]. Table 4.1 shows a review of the most commonly utilized signature datasets.

Table 4.1. Commonly used signature datasets

Dataset Name	Users	Genuine	Forgeries
Kaggle	30	10	10
MCYT-75	75	15	15
Final Dataset	30	5	5

4.2. Pre-processing

As with almost all biometric verification difficulties, the pre-processing performed an essential part in sign confirmation. Below, we review the main pre-processing methods.

Decoding image: in TensorFlow, we could utilize the function (`decode_png`) to decode the picture. So the output was three-dimensional arrays (height, width, num_channels), which we needed for the CNN processing.

Resizing image: this was achieved by (scale) function in TensorFlow. For resizing picture to size utilizing bilinear addition, input pictures could be of diverse varieties, but output pictures permanently floated.

Padding, (tf.pad ()): if the height and width of the picture needed further amount to be in distinct size, padding by zero fed that area.

4.3. Convolutional Siamese Network

A siamese neural network is a type of network structure that normally includes two twin subnetworks, or further if it is used with triplet loss or other metrics that seek more than two inputs. The CNNs hold an identical configuration with similar parameters and distributed weights. Deep CNNs are multi-layered neural networks containing various convolutional layers with varied kernel volumes followed by pooling layers, which review and downsample the output of the convolutions earlier filling near layers. Those subnetworks are connected by a triplet loss function, which computes resemblance metric concerning the far away within the feature representations on all sides of the siamese network [34, 35]. The generic design of the siamese network is described in Figure 4.1 The basic design of a CNN contains three elements, namely (a) convolution layer, (b) pooling layer and (c) fully connected layer. The design of each element is described and explained below, but the fully connected layer not explained, because CNN connected with the one by one kernel and triplet loss.

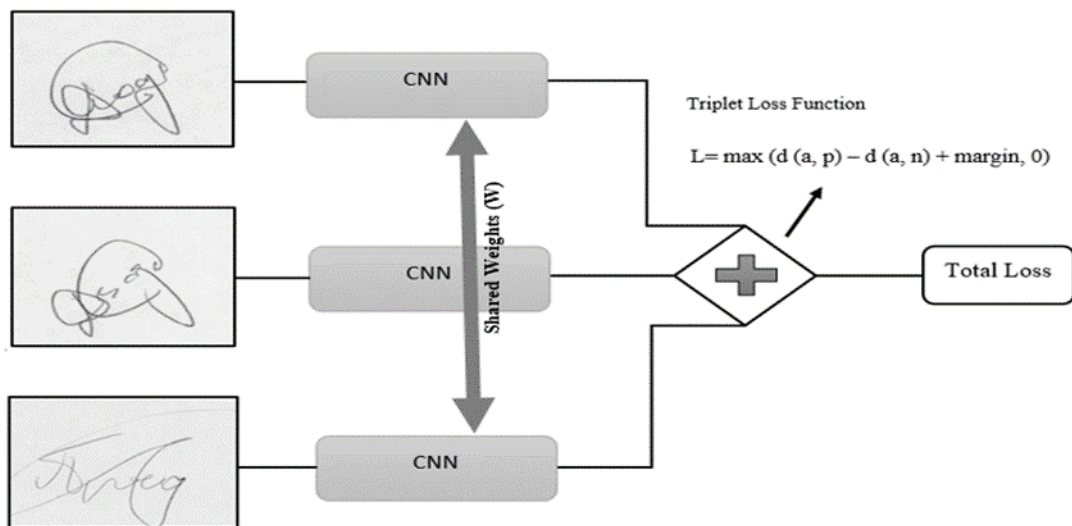


Figure 4.1. Full siamese network architecture [38]

4.3.1. Convolution Layer

Convolution is the initial layer of the educator characteristics of an input picture. Convolution connects within pixels by detecting the characteristics of images utilizing little squares of initial data. It is a scientific method that needs pair records like a model matrix and a filter or kernel. Every picture is able to be declared as a matrix of pixels. A picture from a typical digital camera will produce three layers: red, green, and blue. We could assume those three are two-dimensional matrices above one another, all holding pixel amounts in the area 0 to 255. The kernel applied to the input should be like below.

If the image holds one channel, like a grayscale picture, then a 3×3 kernel is used in $3 \times 3 \times 1$ blocks.

If the model holds three channels, a 3×3 kernel is used in $3 \times 3 \times 3$ blocks.

If the embedding image is a block of characteristic maps of different convolutional or pooling layers and holds a depth of 64, then the 3×3 filter will be used in $3 \times 3 \times 64$ blocks to build the individual amounts to provide the single output feature map.

Figure 4.2 presents a convolution of a 128×128 picture matrix with padding (1) multiplied with a 3×3 kernel matrix and named 'Feature Map' as a production.

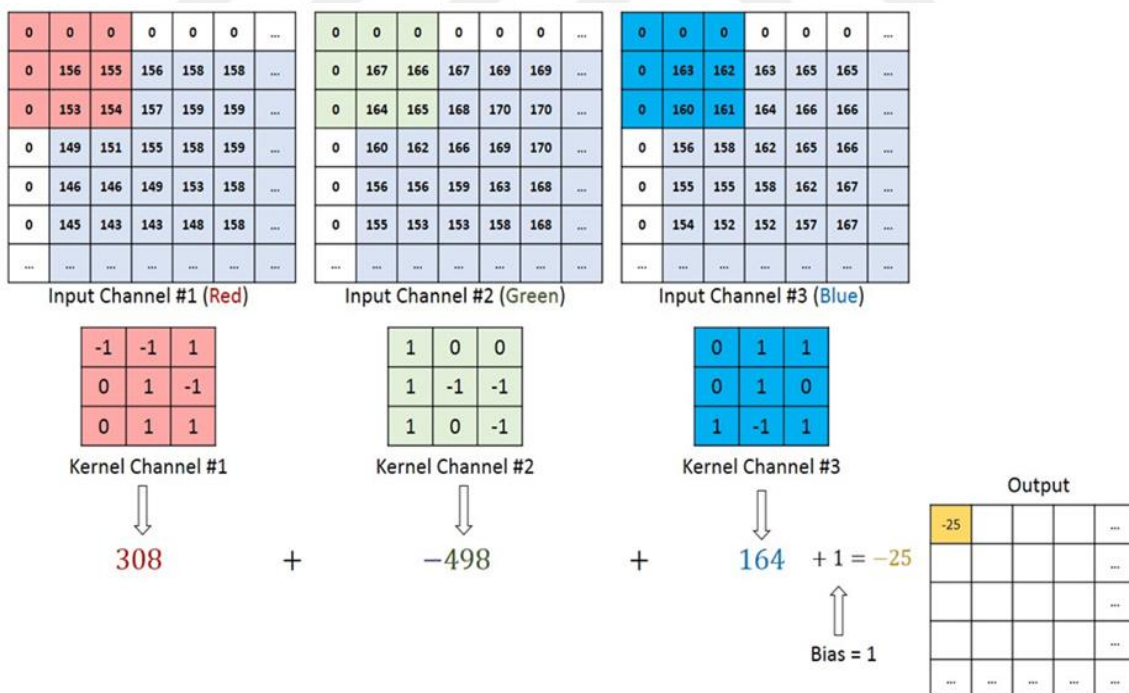


Figure 4.2. Take out a feature map from RGB image [40]

The last output is a layer feature map sized 128×128 reason we used padding 'same', in which sometimes a kernel does not fit the input image. So, in this position, padding by zero is required. The function to determine the future map is:

$$(N + 2P - F) / S + 1 \tag{4.1}$$

In a convolution layer, stride could also be required, which is the amount of pixel change across the initial matrix. When the stride is one, then we crossing the kernel to one pixel to an extent just like in the past model, and whenever the stride is two, the kernel take to two pixels and so on. Rectified linear unit (ReLU) was also used for a non-linear progression. The product is $f(x) = \max(0, x)$. ReLU was necessary because its goal is to propose non-linearity in the ConvNet. The real-world data we needed ConvNet to discover was a non-negative linear amount. Table 4.2 is a summary of the parameter used in the convolution layer to get the best feature map output.

Table 4.2. CNN parameters

Parameter	Value
Padding	1 to filter [3], 2 to filter [5]
Stride	1 or 2
ReLU	$f(x) = \max(0, x)$

One more significant aspect to be cautious with is depth. The gravity of the input or the number of kernels utilized in convolutional layers habitually rises beside the gravity of the network, expanding the quantity of producing feature maps. Further, any network structures, for example, an inception structure, could be too concatenated with the product feature maps of various convolutional layers, which be allowed to enhance the gravity of the input after the convolutional layers. A high number of feature maps in a convolutional network could make an issue as a convolutional process must be made within the gravity of the data. This is a difficulty, particularly if the process implemented is almost high, like as 3×3 , 5×5 or 7×7 pixels, as it could lead to further parameters (weights), thus performing a calculation of the performance (huge space and time difficulty). Grouping layers are sketched to systematically shrink characteristic maps and reduce the width and feature of feature maps in a grid. However, grouping layers does not modify the amount of filters in the form, the gravity, or multiple paths. Deep convolutional networks need similar pooling types that could decrease the width and height of characteristic maps. The resolution is to use a 1×1 kernel to decrease the gravity or number of feature maps. A 1×1 kernel only has one argument or weight for all input channels and is similar to utilising any kernel that appears in a single product purpose. This construction lets the 1×1 kernel play as individual neurons with data within the same location over all of the feature maps in the initial data. This particular neuron could be orderly utilised in a single stride, through left to right and top to bottom, out of the necessity for filling, ending in a characteristic map of the identical width and height as the input. This simple

1×1 kernel gives a way to review data characteristic maps usefully. The method of multiplying filters by 1, in changing, enables adjusting the number of cases of the input feature maps that are created, efficiently letting increasing or decreasing feature maps as required. A convolutional layer by a 1×1 kernel could, hence, be utilized at whatever time in a convolutional network to manage the number of feature maps. As so, it is usually regarded as a projection method or projection layer, or even a feature map or channel pooling layer. The Figure 4.3 below shows how 1×1 works and how to decrease the parameters.

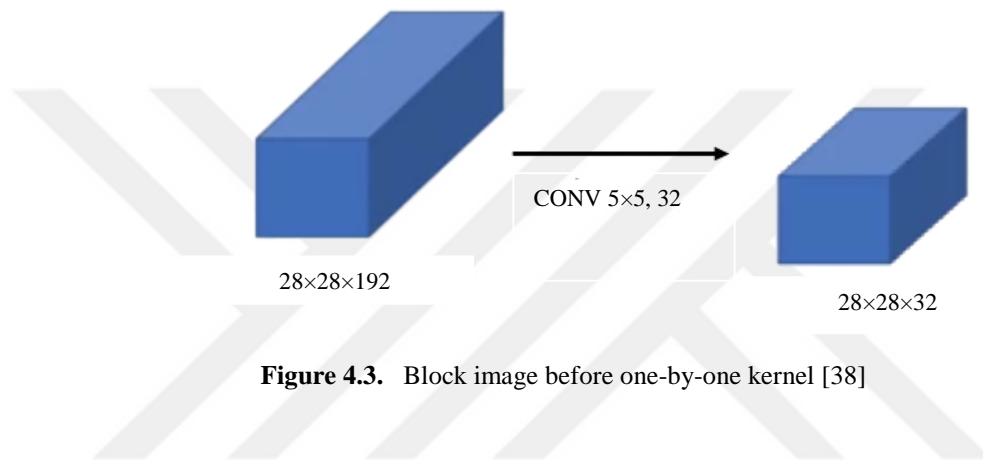


Figure 4.3. Block image before one-by-one kernel [38]

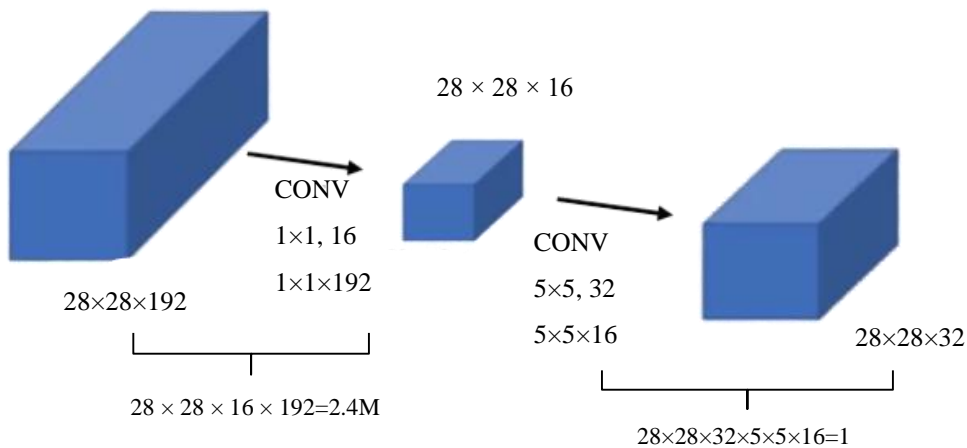


Figure 4.3. (b) Block image with a one-by-one kernel [38]

In Figure 4.3, if we compute the whole feature, we have $28 \times 28 \times 32 \times 5 \times 5 \times 192 = 120M$, which is a large number, but by using a 1×1 kernel, we can reduce this 120M to 12.4M, as shown in Figure 4.3 (b).

The 1×1 kernel allows us to reduce input size data to a lower volume before making the further costly 3×3 , 5×5 or 7×7 convolutions. In this way, it encourages weights to find a more efficient representation of the data. One of the interesting ideas about a 1×1 kernel comes from Yann LeCun: ‘In convolutional networks, there is no such thing as fully connected layers’. There are only convolution layers with a 1×1 kernel and a full-contact table [22, 34].

Yann LeCun says on his *Facebook* page:

“In convolutional networks, there is no such thing as ‘fully connected layers’. There are only warp layers with 1×1 warp beads and a full-contact table. It's a fact that is rarely understood, and that ConvNets do not require a fixed-size input. You can practice them on the inputs that occur to produce a single output vector (with no spatial range), then apply them to bigger images. Rather than a single output vector, you then get a spatial map of the output vector. Each vector sees the input windowpanes in various places on the input. In this situation, ‘fully connected layers’ act as 1×1 manipulation”.

Usually, the 1×1 is a way to reduce the representation dimension (feature maps), such as decreasing the number of colour channels from three to one so it will be reduced if colour is not needed for the prediction. If using a full connection, it will be a more complicated job. In short, the 1×1 requires less weight and a simpler mesh. A full connection requires more training data because the work has additional parameters.

4.3.2. Pooling Layer

The pool section diminishes the size of the argument while the pictures are large. Spatial pooling is also named partial sampling or downsampling, which decreases the dimensions of all maps but keeps significant data. There are two varieties of pool: maximum and average pool. Max pooling delivers the greatest amount of the picture portion concealed by a kernel while average pooling delivers the average of all values of the part of the picture concealed by the filter. Max pooling further acts like noise; it rejects the vociferous activations completely and de-noises using dimensionality compression. However, the average pooling aggregation only reduces the dimensions with noise suppression mechanics. Thus, we could assume that max-pooling is much more useful than average pooling. The most common model is a pooling layer with 2×2 filters applied with a step of two below samples for each slice of input depth of two along both width and height, ignoring 75% of the activation. Each max operation, in this case, would take a maximum of four numbers (small 2×2 region in some depth segments).

Following passing within the above method, we, fortunately, allowed the design to obtain the characteristics. Going forward, we worked to crush the last product and serve it to a regular neural network for analysis goals [22, 35].

4.3.3. Triplet Loss Classification

Triplet loss construction assists us to learn categorized embedding by the concept of identity and dissimilarity. It is a kind of neural network construction where numerous analogous networks are trained and share weights. Through prophecy time, input data is transferred within one network to estimate the grouped embedding's description of input data [22, 35].

The function of triplet loss is described as following:

$$L(a, p, n) = \max(D(a, p) - D(a, n) + \text{margin}) \quad (4.2)$$

The function keeps the far away in the anchor and the positive less than the far away within the anchors and the negative.

As we see in the Figure 4.4, an image is judged by the CNN stages and then displayed as a block of embedding. By adjusting the numbers in the included column and coding the numbers to enforce the triplet loss rule, the original image is now ready. The same method is used to extract embedding's from positive and negative images; minimising it pushes $D(a, p)$ to zero and pushes $D(a, n)$ to be bigger than $D(a, p) + \text{margin}$.

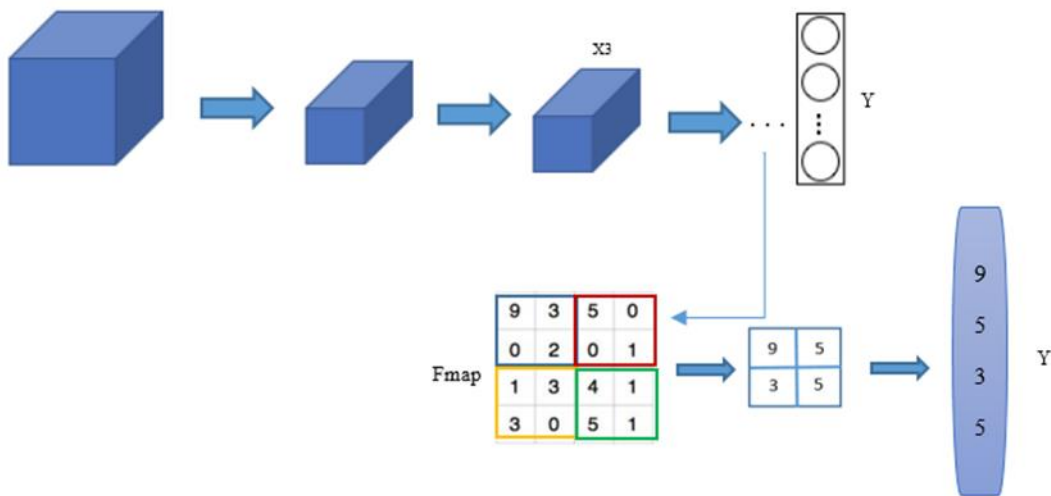


Figure 4.4. Block image displays as a block of embedding [38]

The Figure 4.5 is the full siamese architecture with a CNN for extraction feature maps, using a 1×1 kernel to reduce the depth of the block images. At the end of the CNN, we have an embedding block number; this number includes the information of the images. For instance, they are the line signature; we have three images, and at the end, we must have a three-block embedding number. This number is summated, then the triplet loss function attempts to calculate this number to classify the positive and negative images.

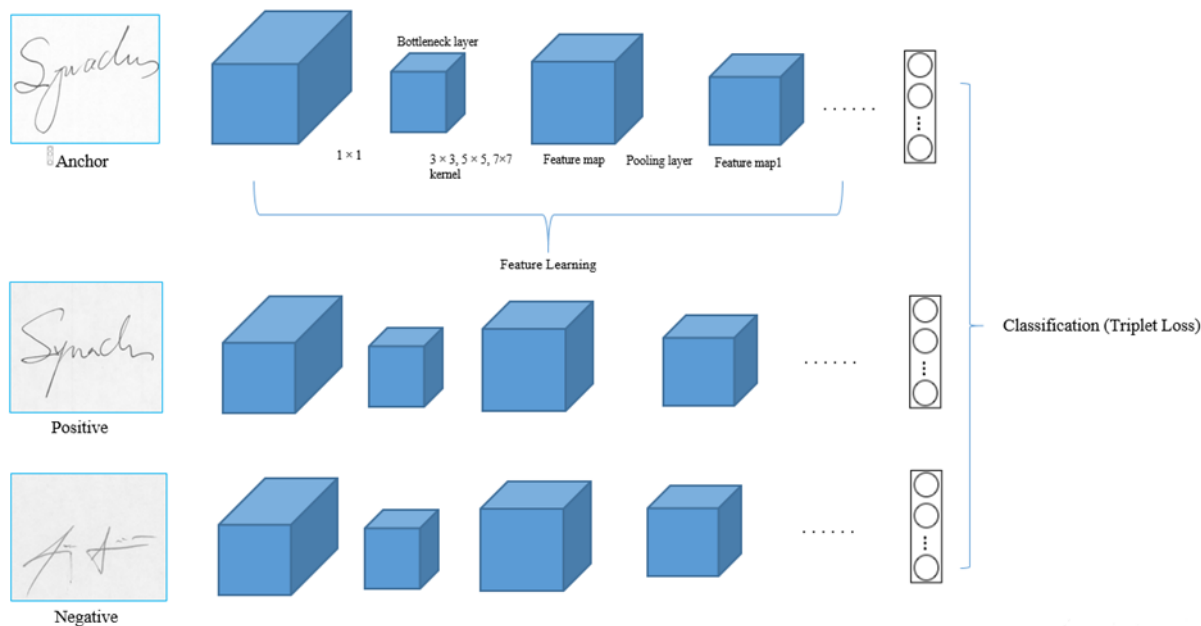


Figure 4.5. Final siamese architecture with classification triplet loss [38]

4.4. Ubuntu operating System

Ubuntu is the number one Linux operating system and the third-most common modern PC operating system. There is an extra 60% of known cloud caseload work on Ubuntu. As the world moves to the Internet of Things, Ubuntu will cover everything from self-driving cars to drones, electronic customers and more [36].

Ubuntu's spread is largely unattributed to its notoriety among software programmers. Exactly what makes the Ubuntu system an attractive development program for such a wide variety of applications — the entire web community, Android application development, ML, data science, robots and self-governing vehicles?

The following section mentions the various significant reasons programmers keep the Ubuntu system.

The premier opportunity for artificial intelligence and ML: Ubuntu is the favored operating system for a host of developing technologies, most reputedly artificial intelligence (AI), ML, and deep learning (DL). For AI, ML and DL, no other operating system matches the depth and breadth of Ubuntu archives, tutorials and cases or implements the equivalent maintenance level for the newest platform variants and available open-source software. This is why Ubuntu is the selected operating system for several of the most common frames, including OpenCV, TensorFlow, Theano and Keras.

Extended assistance from the Ubuntu and the Canonical community: the suitability of other operating system features does not matter if programmers cannot depend on them to be continually stable, secure and up to date. That is why Ubuntu's long-term support releases benefit from five years of Canonical support, including critical bug fixes, shelter updates and hardware empowerment at no cost.

Security: Linux is more secure compared to other operating systems like Windows. Linux is not fully secure due to some malware, but it is less weak than others. Every Linux program, be it an application or a virus, needs administrator authorisation in the form of a password. Unless the password is written, the virus will not be executed. There is no requirement for Linux antivirus software.

Speed: 90% of the world's most active supercomputers work on Linux, compared to 1% on Windows. The computation control of Linux is much greater than Windows, and it begins with excellent hardware support. Windows is not a good leader because it failed to compute speed compared to Linux. Extra features include the use of Docker, which allows one to develop experiences that can work together, conflicting with each other. It assists in building standalone receptacles to manage algorithms, some of which can run quickly only on GPUs and not CPUs. To run Docker receptacles on NVIDIA Docker, which is an NVIDIA GPU, one can use a Linux hostess. For algorithms that speed up a GPU, Linux wins.

4.5. Tensor Flow

The Google Brain venture launched in 2011 to investigate the utility of ultra-deep neural networks, either for study purposes or for use in Google outcomes. As a section of the new product on this plan, the Google Brain Team developed Disbelief in 2011 as the first neural network-based system for DL. The origin code for Disbelief was changed and moved into a library based on a much better application, and in 2015 it became known as Tensor Flow.

Tensor Flow is an interface that easily expresses ML algorithms by creating high-performance neural networks. An account can be dispatched using Tensor Flow, using an extensive variety of systems, extending mobile devices such as tablets and phones to spread policies generally utilised for hundreds of devices and thousands of computing devices, like multi-core CPUs and multi-purpose GPUs.

The principle is adjustable and can be utilised to display an extensive area of algorithms, consisting of practice and conclusion for deep neural network principles. It has continued to be applied to studies and deploys ML systems in products over more than just computer sciences and different areas, consisting of speech recognition, computer vision, robotics, information retrieval and natural language processing [37].

4.6. OpenCV

OpenCV is an open-source archive, built with C and C++ that runs on Linux and Windows systems. There is effective construction on Python, Ruby, MATLAB and other langs. OpenCV is intended to achieve computational performance with an obvious focus on real-time applications. One of OpenCV's aims is to give simplistic, easy-to-utilize computer vision support that assists people immediately create honestly complex vision forms. This archive has over 500 functions covering several fields of vision, consisting of factory outcome inspection, medical imaging, security, user interface and robots. OpenCV further includes a full-purpose machine learning library (MLL). This sub-library focuses on identifying mathematical patterns and aggregation. An MLL is very helpful for the visual businesses that lie at the heart of OpenCV [39].

4.7. Summary

The Table 4.3 below presents a review of the components applied in this dissertation.

Table 4.3. Software and Tools used

Software Component	Name	Version
Operating System	Ubuntu	18.04
Programming Language	Python	3.6.9
Tensor Flow	Tensor Flow	1.14.0
Keras	Keras	2.2.0
Neural Network	Siamese	CSNN
OpenCV	OpenCV	3.4.1
Java Spring Boot	Java Spring Boot	2.1.5
React JavaScript	React JavaScript	16.0
Triplet loss	Triplet loss	Offline

5. RESULTS AND COMPARISON

The purpose of this chapter is to analyze the achieved results, chapter four, displaying the techniques to extracting the FM and the classifier of the system. This chapter shows the results of both train and test sections. Further, it includes a comparison of our outcomes with different subjects that practiced related techniques.

5.1. Training the Model

In all our operations, we practised the CNN with a 1×1 kernel and triplet loss as a classifier. In most practices, we test the model with a learning rate of 0.0001, which we lowered to 0.0000. The model was trained on an Intel (R) Core i7 CPU (installed memory [RAM]: 11.00 GB, Ubuntu: 18.04). The decrease in the loss and increase in accuracy slowed down after 500h of training. We used a Kaggle dataset; the results of the data set are shown in the below table. Table 5.1 shows the result of the Kaggle data set, which contained 30 users with 10 sign signatures each. The test began with a learning rate of 0.0000, and margin α was set to 0.2; the finest outcome of the test was low loss and high accuracy. We test the model with 10 epoch because we obtained the intended outcome as shown in the below table. The training data is in the (train_images) and (train_labels) arrays. To commence training, we request the (model.fit) method—so-called because it "fits" the model to the training data: `model.fit (train_images, train_labels, epochs=10)`.

Table 5.1. Results of the Kaggle dataset

Epoch	Loss	Accuracy
Ep 1/10	0.4973	0.8255
Ep 2/10	0.3743	0.8639
Ep 3/10	0.3345	0.8782
Ep 4/10	0.3128	0.8860
Ep 5/10	0.2937	0.8925
Ep 6/10	0.2815	0.8961
Ep 7/10	0.2673	0.9012
Ep 8/10	0.2573	0.9038
Ep 9/10	0.2470	0.9077
Ep 10/10	0.2393	0.9113

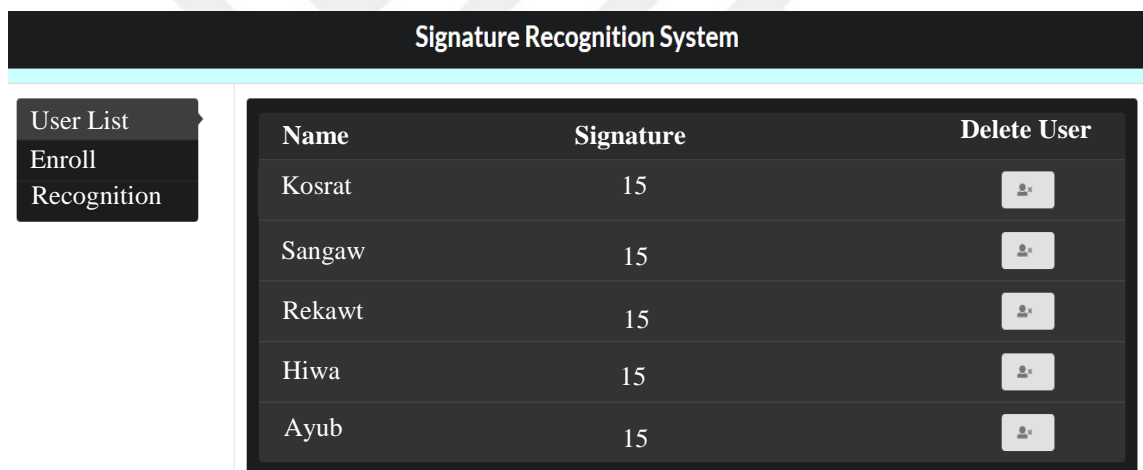
`<tensorflow.python.keras.callbacks.History at 0x7f5ca987cf60>`

As the model trains, the loss and accuracy metrics are presented. This model gives an accuracy of around 0.91 (or 91%) on the training data.

5.2. Testing the Model

We implemented a Spring Boot Java application with a react JavaScript frontend to serve as the inference model. After training the model, we converted the result of the model to a protobuf (.pb) file. In TensorFlow, the (.pb) file contained the graph definition as well as the weights of the model. Generally, we can say (.pb) contained all map features extracted from the images. Thus, a (.pb) file is all that is required to be able to run any given trained model.

There were three parts of the application template. The first was the User List as shown in Figure 5.1; this section showed all registered users and how many sign signatures they had as well as the embedding or the results of the siamese convolutional network, as explained in Figure 4.5. Here, we could remove rejected users.








Name	Signature	Delete User
Kosrat	15	
Sangaw	15	
Rekawt	15	
Hiwa	15	
Ayub	15	

Figure 5.1. User list interface

The second section was Enrolment as shown in Figure 5.2; here, we could insert the employee's signatures to the application, along with their names.

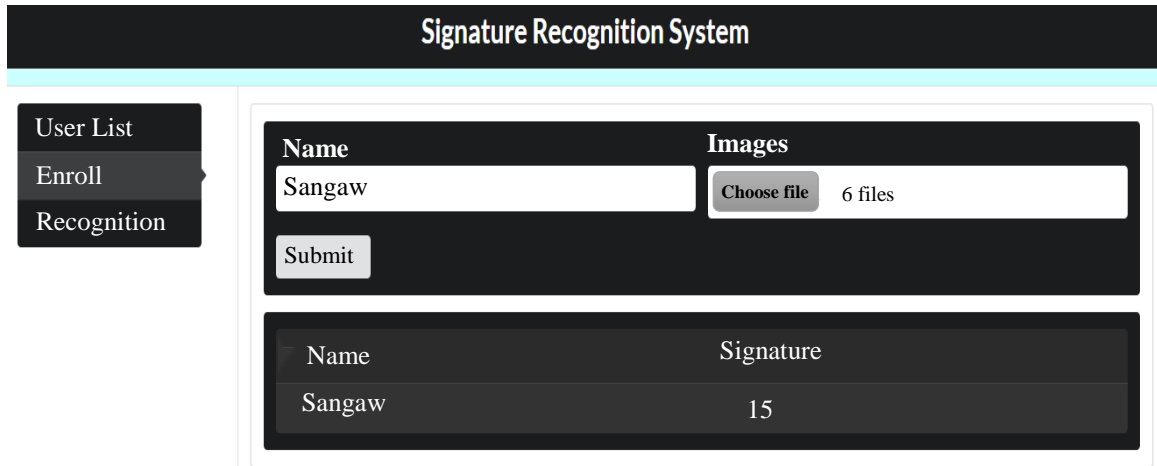


Figure 5.2. Enrolment section to record the users

The third part of the application was Recognise as shown in Figure 5.3; here, we could select the image that we wanted to know was fake or belonged to one of the users that were previously recorded. Also, it showed us how much distance was between the images.

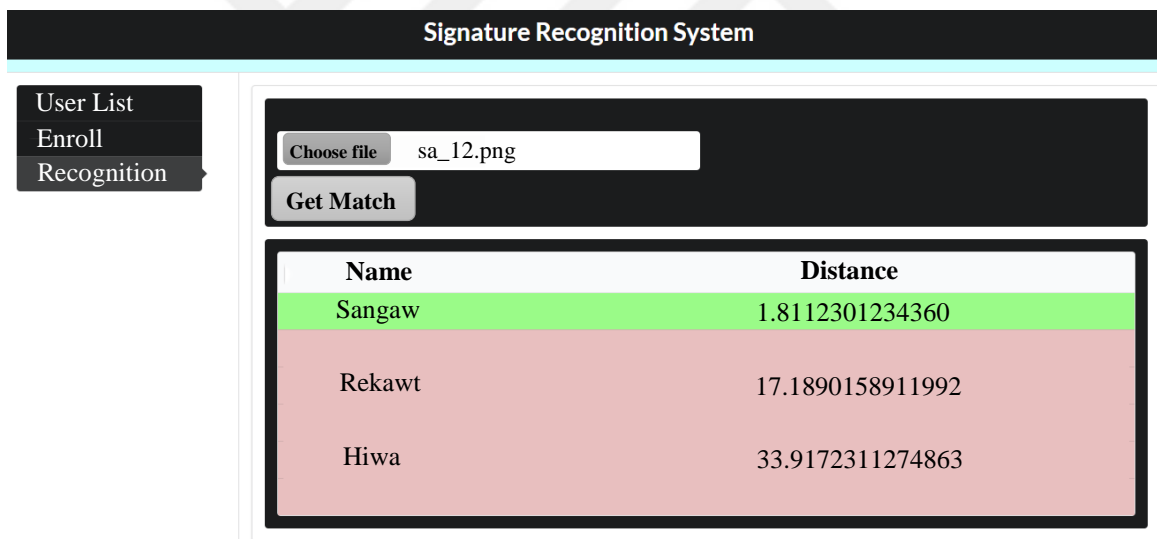


Figure 5.3. Recognise section for recognising the signatures

5.3. Comparison Table

The Table 5.2 below describes the comparison between our system and SigNet model, the reason for choosing SigNet is this model uses the same structure but with different approaches, and the Siemens here has two CNN with the several classifiers as mentions below. There are too many systems in this field of Siemens to confirm the signs but most of them are online and, there is no

need to associate our model with online models because as discussed in the previous chapter that the online models are more performance and more accurate.

Table 5.2. Comparison of approaches

Approaches	SRML	SigNet
Architecture	Sieames net with 3 CNN and one by one kernel without need to using FC layer	Sieames net with 2 CNN and using all layers of CNN (convolutional, pooling, FC)
Classifier	Our model use the Triplet loos function as a cllasifier	This model use euclidean distance function to measure the distance between two signs
Accuracy	Our model just trained on one dataset wich is Kaggle, the accuracy in this dataset was 91%	This model was trainde on several datasets, for instance: the accuracy for CEDAR set was 91% and for GPDS set 88%
Tools and Langs	The major language here is python with tenforflow libraris.	The usage language in this model is python too with using tenforflow libraris
Interface	This model has the interface builded with react java script	There is no interface for this model
System Trainer	The model trained on ubuntu 18.04	The model trained on ubuntu 14
Graphics Card	CPU	GPU

6. CONCLUSIONS AND FUTURE WORK

The goal of this chapter is to give a summary of previous chapters. And discussing the future works on this study in the conclusion section, you will see the detail of our work, the methods, and tools and the network architecture, are mentioned, and the future work section clarifying the look like of the next version of this web application.

6.1. Conclusion

The goal of this activity was to discuss the potency of siamese networks in the area of offline written signature confirmation.

This research performed a complete offline signature verification system, practising a wholly SNN clarification. The system's methodology included data acquirement, pre-processing, feature extraction using CNN, analysis and rating by triplet loss. In this project, we divided the dataset into three parts: anchor pictures, which were the originals; positive images, which were also the originals but were used for testing and comparing to the anchors; and negative images, which were the forged images.

In order to compare the anchor and positive and negative images, triplet loss distance standard was applied. Triplet loss was an excellent option for analysis as it uses the data items and tries to decrease the distance of (a, p) to zero and increase the distance of (a, n) to larger than the distance of (a, p) + margin. All three images went through a CNN model to extract the embedding. A 1×1 convolution was used to decrease the depth of the feature maps, and a pooling layer was used to decrease the width and height of the feature maps. The important point is that the CNN model did not need the fully connected layer because of the 1×1 filter; therefore, we could have a simpler network, using less time and fewer parameters. If the layer is fully connected, it would require more training data because the function has additional parameters.

Therefore, it is clear that triplet loss will provide further scientific outcomes. The test results also explained that the suggested system results in higher accuracy. The system was trained on Ubuntu 18.04 on a CPU using TensorFlow methods. Ubuntu runs faster and is more secure than Windows and can more easily download TensorFlow packages and Python than any other system. With TensorFlow, we could easily implement CNN architecture. To serve the model via an interface, we used a Java application. There were two parts of the application template; the first was Enrolment, which we could use for inserting the employees' signatures to the application, and the second was Recognition, which was used for testing and recognising.

6.2. Future Work

Future work should address the challenges and issues involved in online signature verification, and there is always a scope for new approaches that may improve performance. Future works may include exploring new features and new approaches that may be more effective in distinguishing forgeries from genuine signatures. Also future activity may involve a custom evaluation with a more comprehensive database that includes more than 65 writers. An impressive production would be judging the method with fewer signature examples for each writer so that the method requires fewer examples in the training stage. This technique is called one-shot learning; almost all ML-based object compartmentalisation algorithms need practising on hundreds or thousands of examples and massive datasets. One-shot training tries to get information concerning object classes from one or several training examples. Since the suggested method was examined with random and simple-copy signatures, it would be exciting to examine the method with skilful copies. Other future works may increase the accuracy of the model and improve the features of the application interface and improving the features of the application interface and working more on the react back end.

REFERENCES

- [1] James King, A. I. (2016). A distributed security mechanism for resourceconstrained IoT devices, *Informatica (Slovenia)*, pp. 133-143
- [2] By Michael E. Whitman, H. J. (2014). Principles of Information Security, Cengage Learning, Boston
- [3] Maciej Grzenda, A. I. (2017). Advances in Network Systems: Architectures, Security, and Applications. Switzerland: Advances in Intelligent Systems and Computing, Springer International Publishing Switzerland, pp. 1-10
- [4] Azzopardi, G. (2006). How effective are Radial Basis Function Neural Networks for Offline Handwritten Signature Verification?, University of London, London
- [5] Moubayed, A. &. (2018). E-Learning: Challenges and Research Opportunities Using Machine Learning & Data Analytics, *IEEE Access*, pp. 39117-39138
- [6] Mehryar Mohri, A. R. (2012). Foundations of Machine Learning, The MIT Press, London
- [7] Abbas, A. (2018). Integrated Machine Learning Algorithms for Human Age Estimation, Baghdad University , Baghdad
- [8] Bengio, Y. (2017). Deep learning of representations for unsupervised and transfer learning, *JMLR: Workshop and Conference Proceedings*, pp. 17-37
- [9] Shizhong Han, Z. M. (2017). Optimizing Filter Size in Convolutional Neural Networks for Facial Action Unit, *Computer Vision and Pattern Recognition*, pp. 5070-5078
- [10] Wang, D. L. (2001). Unsupervised Learning: Foundations of Neural Computation., The MIT Press, Cambridge
- [11] Alisher, A. K. (2003). Biometric Identity Verification Using OnLine & Off-Line Signature Verification, MSc Sabanci University, Istanbul
- [12] Diana, K. (2004). Personal Authentication Using Signature Recognition, Lappeenranta University of Technology, Finland
- [13] Luiz G. Hafemann, R. S. (2017). Offline handwritten signature verification — Literature review, *Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pp. 52-60
- [14] A, P. A. (2001). Recognition of Human Signatures, *IJCNN'01. International Joint Conference on Neural Networks*, pp. 1560-1564
- [15] Richard Craze. (2004). Graphology, Hodder & Stoughton House, London
- [16] Gopal Gupta, A. M. (1997). A Review of Dynamic Handwritten Signature Verification, James Cook University Townsville, Australia
- [17] Munich, Mario & Perona, Pietro. (2003). Visual identification by signature tracking, *Pattern Analysis and Machine Intelligence*, pp. 200–217
- [18] Pascual-Gaspara J., Faundez-Zanuy M., Vivarachoc C. (2011). Fast Online signature recognition based on Multi-section VQ, *Elsevier Engineering Applications of Artificial Intelligence*, cilt 24(2), pp. 368–377
- [19] Hemanta Saikia, K. C. (2012). Approaches and Issues in Offline Signature Verification System, *International Journal of Computer Applications*, pp. 45-52
- [20] M. & M. Y. M. H. & M. V. Hanmandlu (2005). Off-line signature verification and forgery detection using fuzzy modeling, *Pattern Recognition*, pp. 341-356

- [21] McCabe A., Trevathan J., Read W. (2008). Neural network-based handwritten signature verification, *Journal of Computers*, cilt 3(8), pp.9-22
- [22] Alexander Hermans, L. B. (2017). In Defense of the Triplet Loss for Person Re-Identification, *Computer Vision and Pattern Recognition*, pp. 1-10
- [23] Toscano-Medina K., Sanchez-Perez G., Nakano-Miyatake M., and PerezMeana H. (2001). On-line signature recognition using feature extraction and multilayer neural networks, *Telecommunications and radio engineering c/c of elektrosviaz' and radio tekhnika*, cilt 56(1), pp.58-70
- [24] Shams. I. Ben. (2007). Signature Recognition by Segmentation and Regular Line Detection, *TENCON 2007 - 2007 IEEE Region 10 Conference*, pp. 1-4
- [25] Porwik P. (2007). The compact three stages method of the signature recognition, *The 6th International Conference on Computer Information Systems and Industrial Management Applications*, pp. 282 – 287
- [26] Bruynooghe, J. B. (2004). A hybrid optoelectronic method for fast off-line handwritten signature, *International Journal on Document Analysis and Recognition*, pp. 56-68
- [27] Ahmad S M S, S. A. (2009). A Hybrid Symbolic-Statistical Approach to Modeling Metabolic Networks, Springer-Verlag Berlin Heidelberg 2007, pp. 132-139
- [28] Shafiei, H. R. (2003). A New On-Line Signature Verification Algorithm Using Variable Length Segmentation and Hidden Markov Models, *Seventh International Conference on Document Analysis and Recognition, ICDAR 200*, pp. 443-446
- [29] Baltzakis H and Papamarkos N. (2001). A new signature verification technique based on a two-staged neural network classifier, *Engineering Applications of Artificial Intelligence 14 (2001)*, pp. 95-103
- [30] P. Deng, H. Y. (1996). Wavelet Based Off-line Handwritten Signature Verification, *Computer Vision and Image Understanding*, pp.173-190
- [31] T. Kaewkongka, K. C. (1999). Off-Line Signature Recognition using parameterized Hough Transform, *Fifth International Symposium on Signal Processing and its Applications, ISSPA*, pp. 22-25
- [32] Xiao, X. &. (2002). Signature verification using a modified Bayesian network, *Pattern Recognition*, pp. 983-995
- [33] Dey, S. &. (2017). SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification, *Pattern Recognition Letters*, pp. 1-7
- [34] Shruti Jadon, A. A. (2019). Siamese Networks for One Shot Learning using Kernel Based Activation functions, *EasyChair Preprint*, pp. 1-12
- [35] Min Lin, Q. C. (2014). Network In Network, *Neural and Evolutionary Computing*, pp. 1-10
- [36] Dickinson, S. (2018). Six reasons why developers choose Ubuntu Desktop, Canonical Company, London
- [37] Abadi, M. &. (2015). TensorFlow : Large-Scale Machine Learning on Heterogeneous Distributed Systems, Google Research, California
- [38] S. Mshir and M. Kaya. (2020). Signature Recognition Using Machine Learning. *2020 8th International Symposium on Digital Forensics and Security (ISDFS)*, pp. 1-4
- [39] Adrian Kaehler, G. B. (2017). Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library, O'Reilly Media - Technology and Business Training, California

- [40] P. Patel, "Medium," Evan William, 15 August 2012. [Online]. Available: <https://medium.com/@priyankapatel2205/convolutional-neural-nets-1813eee0510>. [Accessed 14 August 2020]



CURRICULUM VITAE

Name SURNAME

PERSONAL INFORMATIONS

Birth of Place : Arbil - Iraq
Birth of Date : Jun 11, 1995
Nationalty : Iraqi
Address : Erbil - Iraq
E-mail : Shalaw.00115594@gmail.com
Languages : English (B2), Arabic (B2)

RESEARCHER INFORMATION

Student Orcid ID : <https://orcid.org/0000-0002-5421-9038>
Advisor Orcid ID : <https://orcid.org/0000-0003-2659-3601>

EDUCATION

Bachelor : University of Koye, Faculty Engineering, Department of Software, Year 2013-2017
High School : Koye kwran High School., Koye City, Year 2010-2013

RESEARCH EXPERIENCES

- ✓ Database Technologies (Ms access, Oracle, Sql server, Sql)
- ✓ Web Technologies (HTML5, CSS, PHP, Java Script)
- ✓ Mobile Application (Flutter, Java, Sqlit)

ACADEMIC ACTIVITIES

Paper:
Mshir, Shalaw & Varol, Prof. (2019). A New Model for Creating Layer Planes Using Steganography for Text Hiding. 1-5