

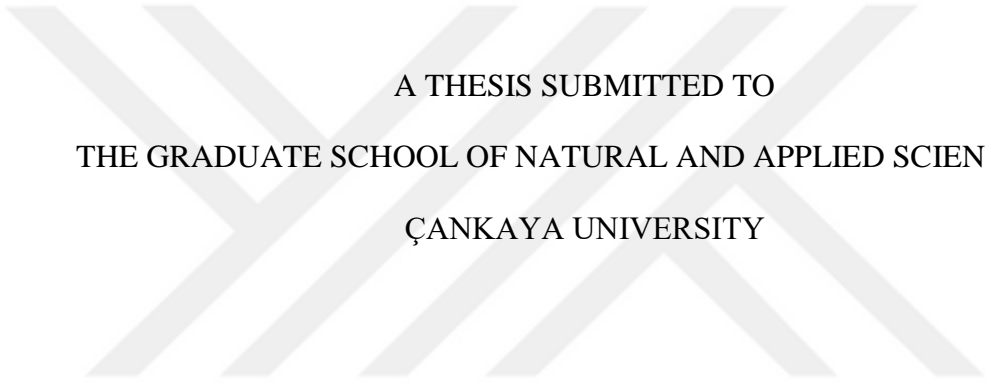


**MATCHING COMPOSITE DRAWINGS AND MUGSHOT PHOTOGRAPHS
TO DETERMINE THE IDENTITY OF THE PERSON**

MUSTAFA KARASOLAK

APRIL 2019

MATCHING COMPOSITE DRAWINGS AND MUGSHOT PHOTOGRAPHS TO
DETERMINE THE IDENTITY OF THE PERSON



A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES OF
ÇANKAYA UNIVERSITY

BY

MUSTAFA KARASOLAK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF

MASTER OF SCIENCE IN
COMPUTER ENGINEERING DEPARTMENT

APRIL 2019

Title of the Thesis: **Matching Composite Drawings and Mugshot Photographs to Determine the Identity of the Person.**

Submitted by **Mustafa KARASOLAK**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.



Prof. Dr. Can ÇOĞUN

Director


I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.



Prof. Dr. Erdoğan DOĞDU

Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.



Roya CHOUPANI

Supervisor

Examination Date: 30.04.2019

Examining Committee Members

Asst. Prof. Dr. Roya CHOUPANI

Asst. Prof. Dr. Yuriy ALYEKSYEYENKOV

Asst. Prof. Dr. Abdül Kadir GÖRÜR

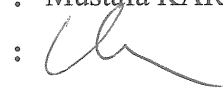


STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Mustafa KARASOLAK

Signature :



Date

: 13.05.2019

ABSTRACT

MATCHING COMPOSITE DRAWINGS AND MUGSHOT PHOTOGRAPHS TO DETERMINE THE IDENTITY OF THE PERSON

KARASOLAK, Mustafa

M.Sc., Department of Computer Engineering

Supervisor: Roya CHOUPANI

April 2019, 73 pages

In this thesis, a new photo-sketch generation and recognition technique is proposed using residual convolutional neural network architecture. For this, the proposed architecture is trained with face photos and sketches. Sketches are applied to the proposed Region-based Convolutional Neural Networks (RCNN) architecture and, face photos are obtained at network output. Then, the obtained face photographs are compared with the images in the database. It is associated with the highest similarity photograph. Structural Similarity Index (SSIM) is used to measure similarity. It is very useful for law enforcement for image processing applications. 188 images are used for training and testing. Of these, 148 are used for training. 20 are used for validation and 20 are used for testing. Data augmentation is applied to 148 images used for training. As a result of the data augmentation process, 444 face images are obtained and used for network training. As a result of network training, the success of the training curve is 90.55% and the validation success is 91.1%. True face recognition success from generated face images with SSIM is 93.89% for CUHK database and 84.55% AR database.

Keywords: Face detection, Residual CNN, Convolutional Neural Networks, Mugshot Photographs.

ÖZ

KİŞİ KİMLİĞİNİN BELİRLENMESİNDE KOMPOZİT ÇİZİMLER VE MUGSHOT GÖRÜNTÜLERİNİN EŞLEŞTİRİLMESİ

KARASOLAK, Mustafa

Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı

Tez Yöneticisi: Roya CHOUPANI

Nisan 2019, 73 sayfa

Bu tezde, residual konvolüsyonel sinir ağ mimarisi kullanılarak yeni bir fotoğraf taslak oluşturma ve tanıma tekniği önerilmiştir. Bunun için önerilen mimari yüz fotoğrafları ve el çizimleri ile eğitilmiştir. El çizim görüntüleri, önerilen Region-based Convolutional Neural Networks (RCNN) mimarisine girdi olarak uygulanır. Daha sonra, elde edilen yüz fotoğrafları very tabanındaki görüntüler ile karşılaştırılmıştır. En yüksek benzerlik oranına göre görüntüler ilişkilendirilmiştir. Benzerliği ölçmek için Yapısal Benzerlik Endeksi (Structural Similarity Index-SSIM) kullanılmıştır. Görüntü işleme uygulamaları kapsamında önerilen yöntem güvenlik güçleri için oldukça yararlı olabilir. Eğitim ve test için 188 resim kullanılmıştır. Bu görüntülerden 148 tanesi eğitim, 20 tanesi doğrulama, 20 tanesi ise test için kullanılmıştır. Veri artırma yöntemleri, eğitim aşaması için kullanılan 148 görüntüye uygulanmıştır. Veri artırma sürecinin bir sonucu olarak, 444 yüz resmi elde edilmiş ve ağ eğitimi için kullanılmıştır. Ağ eğitimi tamamlandıktan sonra, eğitim eğrisinin başarısı % 90.55 ve doğrulama başarısı % 91,1'dir. SSIM ile oluşturulan yüz görüntülerinden elde edilen gerçek yüz tanıma CUHK veri seti başarısı % 93.89 ve AR veri seti başarısı % 84.55'dur.

Keywords: Yüz Tanıma, Residual Konvolüsyonel Sinir Ağları, Konvolüsyonel Sinir Ağları, Mugshot Görüntüleri.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Asst. Prof. Dr. Roya CHOUPANI for his supervision, special guidance, suggestions, and encouragement through the development of this thesis.

It is a pleasure to express my special thanks to my family for their valuable support.



TABLE OF CONTENTS

	<u>Page</u>
STATEMENT OF NON-PLAGIARISM PAGE	iii
ABSTRACT	iv
ÖZ	v
ACKNOWLEDGEMENTS	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	ix
LIST OF ABBREVIATIONS	x
CHAPTER 1	1
INTRODUCTION	1
1.1 Background	1
1.2 Objectives.....	2
1.3 Organization of the Thesis	3
CHAPTER 2	4
CURRENT BASED TECHNIQUES	4
Fundamental Concept of the Current Based Techniques.....	4
2.1 Forensic Sketch Matching.....	5
2.1.1 Detection of Forensic Faces in Single Image.....	6
2.1.2 Adversaries Machine Learning	6
2.1.3 Manual Facial Comparison from Forensic Expert	8
2.2 Matching Composite Sketch with Mugshots	8
2.3 Recent Studies in Facial Recognition	11
2.4 Matching Forensics Sketches with Mugshots	14
2.5 Image Retrieval of Face Image on the basis of Probe Sketch with Sift Feature Descriptors	15
2.6 Matching Composite Sketches with Face Photographs: Component Base Approach.....	16
2.7 Matching Face and Retrieval for Forensics Applications	17
CHAPTER 3	20
MATERIAL AND METHOD	20
3.1 Artificial Neural Networks.....	20
3.1.1 Single Neuron Model	20
3.1.2 Activation Functions	21
3.1.3 Feed Forward Networks.....	23
3.1.4 Back Forward Networks	25
3.2 Deep Learning.....	26
3.2.1 Image Classification with Deep Learning.....	27
3.2.2 Convolutional Neural Networks	28
3.2.3 Learning Process	31
3.2.4 Selection of Data Sets	32
3.2.5 Cost Functions.....	33

3.2.6 Gradient Descent.....	35
3.2.7 Momentum.....	36
3.2.8 Stochastic Gradient Descent	37
3.2.9 Back propagation	38
3.2.10 Regularization Technics.....	41
3.3 Data Description.....	43
CHAPTER 4	45
PROPOSED METHOD.....	45
CHAPTER 5	50
RESULTS	50
CHAPTER 6	56
CONCLUSION.....	56
APPENDICES	57
REFERENCES.....	69

LIST OF FIGURES

	<u>Page</u>
Figure 1 (A) Sample Passport Photographs (B) Hand Drawn Sketches	2
Figure 2 Creating composite sketch with Identikit	9
Figure 3 Calculation of neurons	21
Figure 4 Graph of ReLU activation function	22
Figure 5 Graph of sigmoid activation function	22
Figure 6 Graph of Tanh activation function.....	23
Figure 7 Single Layer Perceptron Model	24
Figure 8 Multi-Layer Perceptrons Model	25
Figure 9 CNN stages	28
Figure 10 Access to saturation point.....	36
Figure 11 Local and Global Minimum.....	37
Figure 12 Simple chain rule	38
Figure 13 Back propagation on neural network.....	39
Figure 14 Weights Selection	42
Figure 15 a) Conventional Neural Network b) Dropout Application	43
Figure 16 Image Dataset. a) face images b) sketch images	43
Figure 17 Sketch-Face recognition problem	45
Figure 18 Proposed Method	46
Figure 19 Image Dataset. a) original images, b) gaussian noise, c) salt-papper noise	47
Figure 20 Training and validation curves of proposed CNN structure	51
Figure 21 Face creation results of the proposed CNN architecture, a) sketch images, b) face image results of proposed CNN structures, c) ground truth face photos in CUHF dataset.....	52
Figure 22 Face creation results of the proposed CNN architecture, a) sketch images, b) face image results of proposed CNN structures, c) ground truth face photos in AR dataset.....	53
Figure 23 Face SSIM curve.....	54
Figure 24 Face correlation curve.....	55

LIST OF ABBREVIATIONS

ANN	: Artificial Neural Networks
ASM	: Active Shape Model
CE	: Cross Entropy
CNN	: Convolutional Neural Networks
CUFS	: CUHK Face Sketch
E-HMM	: Embedded hidden Markov model
LDA	: Linear Discriminant Analysis
LFDA	: Local Feature-Based Discriminant Analysis
LLE	: Local Linear Embedding
MLBP	: Multiscale Local Binary Patterns
PCA	: Principal Component Analysis
RCNN	: Region-based Convolutional Neural Networks
ReLU	: Rectified Linear Unit
SSIM	: Structural Similarity Index-Measurement

CHAPTER 1

INTRODUCTION

1.1 Background

Matching sketches and mugshots is considered to be the most complicated and challenging tasks in computer visualization field due to intra-class changes caused by facial look variations, expression, and illumination. The process becomes very complex and non-linear in all the space dimensions that are linear to provide image space dimension. The problem of robust face detection thus becomes more complex. Face detection is frequently applied in criminal investigations. With the visual biometric technology continually advancing, new technologies are continuously improving the matching of composite drawings to facial photographs. There are several methods of matching hand drawn sketches to photographs but very little research has been done on the matching of computer generated composite photographs to passport photographs. The use of automated face recognition algorithms matched with computer generated composites has not been extensively studied. The challenge involved with face detection of composite sketches is that computer generated sketches usually look stilted while most hand drawn sketches looks raw due to the fact that the artist is able to add a psychological edge to the drawings creating an exact facial component shape and even shading. This is the reason why it is easier to recognize a hand-drawn sketch compared to the composite sketch. In Figure 1, there are samples of Passport photographs, Hand drawn sketches and composite sketches

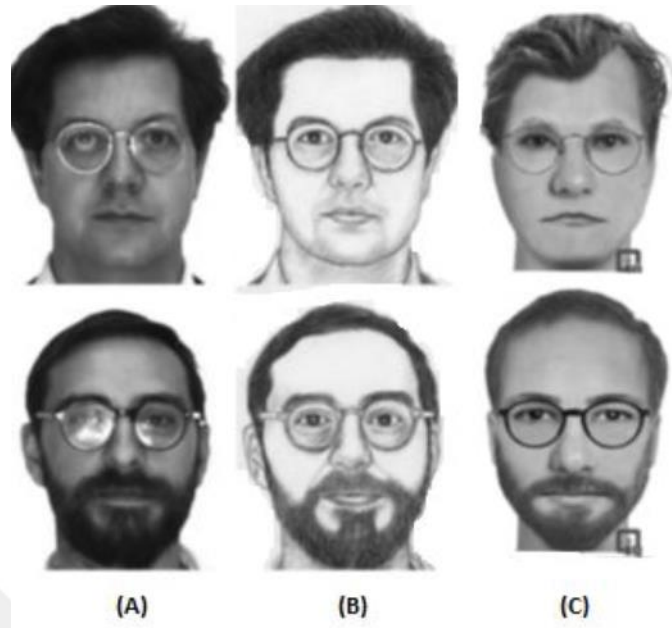


Figure 1 (A) Sample Passport Photographs (B) Hand Drawn Sketches
(C) Composite Sketch

1.2 Objectives

In this thesis, a highly effective method is proposed for the face-sketch recognition task which is crucial for several law enforcement agencies. The aim of this work, which allows real face photographs to be obtained using Sketch, is to identify real face photographs of possible offenders and thus capture them more efficiently. For this purpose, sketch features are automatically obtained by Region-based Convolutional Neural Networks (RCNN). The RCNN features are then converted to their original dimensions using deconvolutional layers. During training, the images obtained at the network output are compared with the ground truth images and the network is updated. In the test phase, the images obtained at the network output are compared with the images stored in the data set. In this process, target image is matched with an image with the greatest SSIM. Images that remain under a certain SSIM value are not matched with any images in the database. The CUHK Face Sketch Database (CUFS) is used for training and testing of the proposed convolutional neural network architecture.

1.3 Organization of the Thesis

This thesis contains five chapters. All the necessary information about matching composite drawings, mugshot photographs and methods used for image generation can be constructed with deep learning methods.

Chapter 1 is an introduction to the history and objectives of this thesis.

Chapter 2 includes an introduction of composite drawings, mugshot photographs and region based convolutional neural network which will be used in this thesis. In addition, previous matching techniques are introduced.

In Chapter 3, material and methodology are studied.

In Chapter 4, proposed method is explained

In Chapter 5, obtained results are represented.

Finally, chapter 6 contains the conclusion.

CHAPTER 2

CURRENT BASED TECHNIQUES

Fundamental Concept of the Current Based Techniques

This chapter aims to identify, highlight and compare technical composite drawings to passport photographs in order to determine the identity of said person based theoretical understanding of the phenomenon. In this chapter, a clear and precise understanding of the dynamics of the identity of the individual will be analysed through different tools and techniques to compare those composite drawings and passport size photographs. In addition to that, algorithms such as facial recognition algorithms will be analysed and examined to illustrate the method of comparing and examining the person's identity. All these tools are very important and reflect the enhancement of the technology and how such advancement has helped the investigative agencies and other relevant authorities in identifying the person's identity.

For that matter, this chapter will primarily shed light on the forensic sketch matching which is a tool for investigators when there is no video from the crime scene available. For that matter, the other means of analysis such as mugshot analysis along with other factors are examined to understand the identity of the individual. In addition to that, adversary machine learning is another factor that will be discussed in this chapter in detail. The purpose of such addition in this chapter is to help in theorizing the overall mechanism of face detection through machines. In due course, learning how to use a machine for any particular phenomenon will be analysed and examined in this chapter. Also, the importance of working groups and how to do manual facial comparisons will be discussed in this chapter.

Identification of a particular face cannot only be done with a click rather there is a process that has to be followed in this regard. This chapter will shed light on the several aspects and technicalities that involve matching composite sketching, facial

recognition algorithms, matching forensic sketches and retrieval of face image on the basis of probe sketch with sift feature description. In addition to that, another technique that is used to detect a particular face is the matching of composite sketches with face photographs which is known as the component based approach. There are other applications that are involved in the identification of faces. The basic framework that is being used by the experts is the Viola and Jones framework and this framework will be explained and utilized in this regard for better explanation of the given concepts and tools to identify the facial sketch.

2.1 Forensic Sketch Matching

One major task carried out by forensic examiners at the time of investigation is forensic sketch matching which is done when no image or video from the crime scene is available. Forensic examiners carry out manual examination on facial videos or images for matching with huge database that comprises of mugshots. Automated system usage for facial recognition not just increases efficiency of investigation carried out by forensic workers at agencies but also standardizes process of comparison [1]. Although system of biometric face recognition has been used in securing access of building, Civil ID, border control & verification of user login, however such system is not present that can be used in identification or verification within crime investigation like comparing images taken from CCTV with available mugshots database.

The consequence of wrong decision which forensic face recognition make can be more severe than biometric recognition. Reason is the large unevenness that exists in faces like lighting conditions, pose, and facial expression and also in systems of imaging itself like quality of image, compression and resolution [1-2]. It is also vital to mention that in forensic cases quality of image that is available to be processed is mostly low like images captured at crime scene through CCTV [3]. These images often times have low resolution, restricted pose and sometime have half blocked faces. However, task of recognition in forensic is “offline” in comparison with biometric system in which decision must be taken in real time, for example system in case of building or border access. The literature reviews research over forensic sketch matching.

2.1.1 Detection of Forensic Faces in Single Image

Methods of detection of single image can be divided in four categories. Few methods overlap the boundary of classification as discussion in few papers [4-6].

Knowledge-based methods: These methods involve coding that includes typical face, for example relationship among facial features. This method is rule-based that encodes face features of human. Usually relationship among features is captured by the rules. Such methods mainly are designed for the purpose of localization of face.

Feature Invariant Approaches: Principal aim in the algorithm is finding structural features which are present even when viewpoint, pose or conditions of lightning alter & these are used for locating faces. The methods are also mainly designed for localization of face.

Template Matching Methods: A number of standard face patterns are stored for describing the whole face or few facial features independently. The relations between stored patterns and input image are determined for detection. Such methods are used for both localization of face & its detection.

Appearance based Methods: Appearance-based methods differ from template matching in that the templates or models are learned from training images' set which should record representative variability present in face appearance. Such models are afterwards used in detection. Such methods are mainly designed to detect face.

2.1.2 Adversaries Machine Learning

Machine Learning is a method of data analysis that automates the development of analytical models. Using algorithms that learn interactively from data, machine learning allows computers to find hidden insights without being explicitly programmed to look for something specific. The purpose of adding machine learning in this context is because face detection is all about using the machine and tools in order to accurately and adequately examine and enhance the dynamics of the face. For that, knowing a machine is pivotally important in order to accurately detect a machine.

Because of the new computing technologies, today's machine learning is not like the machine learning of the past. While many machine learning algorithms have been around for a long time, the ability to automatically apply complex mathematical calculations to the big data-increasingly and faster-is a recent development. Machine

learning is one of the sub-branches of artificial intelligence. In particular, it is seen that some of the techniques in machine learning have more ability to learn. It has become even more widespread with the high accuracy and accuracy of the problems. Here are some widely publicized examples of machine learning applications that one may already be familiar with:

Interest in machine learning has resurfaced due to the same factors that have made data mining and Bayesian analysis more popular than ever. Things like the increasing volume and variety of data available, computational processing that is cheaper and more powerful and the storage of data in an accessible manner. All of this means that one can quickly and automatically produce models that allow one to analyse larger and more complex data and deliver faster, more accurate results - even on a very large scale. The result is that high value forecasts that can lead to better intelligent decisions and actions in real time without human intervention.

A secret to the production of intelligent actions in real time is the development of automated model. The leader of innovative analytics ideas Thomas H. Davenport wrote in The Wall Street Journal that, with rapidly evolving and growing data volumes, one need rapidly evolving modelling flows to be able to go along". And one can get this with machine learning. According to him, "people can usually create one or two good models per week; Machine learning can create thousands of models per week.

It is difficult to briefly define what AI is. In the famous book Artificial Intelligence, used as a basic bibliography in several universities, the authors spend 30 pages discussing the subject. There are several different AI approaches to solving a wide variety of problems. One of these approaches is the machine learning, one of the most relevant areas within the AI and that will be analysed next. Machine learning algorithms look for patterns within a set of data. These algorithms have been around for a long time, but there has never been such a large amount of digital data available to feed these algorithms as today thanks to two factors: mass computerization and the emergence of the Internet.

In the physical media era, the possibilities for automatic recommendation were quite limited: your CD / tape / LP library, for example, was inaccessible to algorithms. Amazon could use its purchase of discs and compare it with other people, but all the discs you earned as a gift or bought elsewhere would still be inaccessible to it. The

digital sale of music has increased the possibilities, so much that Apple tried to emulate the Genius and even the Ping social network in the iTunes Store. It did not work, but not for lack of raw data. So it was the streaming services that actually fulfilled the promise of machine learning applied to music. These services know all your libraries, playlists and who you follow. Most importantly, they have this information from several other users, and can compare their behaviour with that of others. Many other areas had this boom of information available as the songs, which increased the interest of academia and the market in machine learning techniques.

2.1.3 Manual Facial Comparison from Forensic Expert

This section gives brief overview about how forensic expert makes facial comparisons. The discussion is based upon guidelines which are set by workgroup on comparison of face at “National Institute of criminology and forensic science (NICFS)” [9]. Face comparison is based upon morphological anthropological features. Many times, it is hoped that pictures are obtained in similar pose. Comparison is mostly focused upon: shape of eyes, mouth, ears, nose etc, distance between various relevant features, contour of chin & cheek lines, wrinkles, scars & moles etc. on face.

When manual comparison of face is being made, it should be considered that differences might invisible because of overexposure, underexposure, too low resolution, distortion and out-of-focus in imaging. Moreover, feature that are similar can result to distinct depictions because of camera’s position in comparison with head, inadequate resolution, difference between focus of two images & disturbance in the process of imaging. Because of these effects that mostly make process of comparison hard, anthropological face features are compared visually and termed as: different, no observation, different in details, similar, similar in details. Apparent differences and similarities are evaluated further classification as: Strongly discriminating, moderately discriminating, weakly discriminating. There is strong requirement of automating process as it will not just increase speed with which comparison takes place but will also help in standardizing process.

2.2 Matching Composite Sketch with Mugshots

Procession within biostatistics upgrades the law enforcement agencies through offering means of identifying criminals in short time. Visual Biometrics in

combination with Fingerprint recognition offers face recognition that provides facial features analysis for recognizing identity of an individual. In several scenarios, suspect's facial photograph is not available. Under such circumstances, drawing sketch complying description that is provided from a victim or spectator is a method which is used commonly for assisting the police in identification of possible suspects. For drawing forensics sketches the artists in the police force requires excessive training in pictorialization and drawing. On the hand compose sketching requires couple of hours for training that enables even non-artist to compose sketch by using software of composite sketching which becomes perfect alternative to offer assistance with investigation. The usage of composite sketching is beneficial as it consumes less time and is more economical. Some of the facial composite software kits that are most widely used are EvoFIT, Photo-Fit, IdentiKit, Mac-a-Mug and FACES. Some are used for detecting principal Facial Components such as Eyebrow, Hair, nose, eyes, shape, mouth, eyeglasses, etc. and subaltern Facial Components like moles, tattoo, smile lines and scars etc. Figure 2 highlights procedure of creating composite sketch with the help of Identikit. In this system of facial composition, facial components are chosen from candidate list shown upon a system Graphical User Interface. The difference that exists between forensic sketches (sketches drawn by hand) and computer generated sketches is evident from Figure 2. Compared with face photos (mug-shots), both composite sketches and both hand drawn sketches lack comprehensive texture specifically around cheeks and forehead.



Figure 2 Creating composite sketch with Identikit

However, artists are able to depict every facial component having exact same shape and shading. Thus, the sketches drawn by the artists capture the very distinctive

features present in various faces. In contrast, the facial components within composite sketches are required to be approximated by most similar components that are present in the database of the composite software. Furthermore, artist's psychological mechanism guarantees that the sketches which are drawn from hand appear raw while composite might appear stilted. Therefore, while person may be easily recognized with the help of hand-drawn sketch, it is many times a challenge to recognize a person from his composite sketch. Observations similar to this have been reported in community of cognitive psychology where survey revealed that 80 percent officers of law enforcement agencies used sketches generated by computer software [10]. Despite of the high number of law agencies utilizing computer generated composite sketches, application of automated facial recognition algorithms in computer generated composites has not been studied sufficiently. In contrast, more attention has been given to both forensic sketches and viewed sketches.

Research on matching of sketches only started since a decennium. Due to inaccessibility of standard public database of forensics sketches, most research carried out on viewed sketches is from past 10 years. Much of the early that focused upon viewed sketches is of Tang et al. [11-12]. Synthetic photograph is produced out of sketch in these researchers & match is performed through standard algorithms of face recognition.

A lot of problem persist for forensic sketch comparison with normal facial recognition in which both gallery & probe images are photographs. Fineness in sketches whether they are forensic or viewed sketches are different from large mug-shot gallery. Most of the work previously done focused upon viewed sketches of forensics sketches. Additional problems are presented in forensic sketches in comparison to viewed sketches. Due to memory's fractious nature, spectator is not able to remember exact visual aspect of the criminal. This leads to inaccurate and incomplete depiction of sketches that considerably reduces the performance of recognition.

For handling such difficulties, local feature-based discriminant analysis (LFDA) was developed by Klare et al. [13] which is able to learn discriminative representation out of partitioned vector of LBP and SIFT features by use of multiple discriminative subspace projections [14-15]. Component-based facial recognition means were studied by a number of researchers [16-20]. However, indirectly or directly utilized features

of intensity that show sensitivity with respect to alterations of facial features or adopted supervised algorithm for classifying performance of whose is delicate towards quantity data of training available. Moreover, proposal of such algorithms was mostly for resolving problem of misalignment of photo-to-photo face matching & does not address heterogeneous modality break that exists during matching of composite sketches generated by computer with facial photographs.

2.3 Recent Studies in Facial Recognition

The history of algorithms of facial recognition which are able to generate completely automatic output rests upon many studies which can be regarded as pioneers in the field and serve as base line for several studies to follow. It is based upon comparison first proposed in the early 1960s and done through manually tagging of the characteristics features like nose, mouth, ear and eye on input picture, then determining distance which of such features with reference point and comparing results with other data. In the 1970s, in approach which involved choosing twenty-one different subjective markers like lip thickness and hair colour, the mentioned measurements and calculations were still carried out manually [21]. An Eigenface approach (PCA-Principal Component Analysis) was put forward in 1988 that served as basis for further studies to follow. For the first time in this approach, a normalized eigenface was developed by use of limited values [22]. Identifiers and calculations were applied over the eigenface that is made of whole face's specific vector area. In the year 1991, while dealing with work over eigenfaces, through making use of residual errors, the possibility of identification of faces from bigger and more general images became reality and it thus became precursor of automatic and more reliable form of facial recognition system [23].

Most progress in matching of viewed sketches has been carried out by Tang et al. [24]. Eigen transformation method was first used by Tang and Wang for either project sketch image in photo subspace or for projecting image of photo in sketch subspace. After projection within same subspace of image, they were matched by use of PCA-based matcher [11]. Tang and Wang carried out similar transformation based on eigen; however transformation was applied separately on texture and shape of face. Liu et al. [12] described method of synthetic sketch generation which would convert image of

photo in a geometry while preserve the synthetic sketch through Local Linear Embedding (LLE). Wang and Tang [5] offered an improvement in this method and modelled relationship between photo image and sketch patches with Markov random field. For the purpose of minimizing energy between image selected and their photo mates or corresponding sketches as well as their selected neighbour patches belief propagation was used. In research of Liu et al. [12] and Wang and Tang [5], the generated synthetic sketches were compared with gallery of photo images by use of different standard algorithms of face recognition.

Lin and Tang [18] provided generalized method of heterogeneous face biometrics which was applied among photo modalities and sketch (as well as visible spectra and NIR). In matching photos and sketches they developed an understand of linear transformation method which was able to project images from two domain in a single feature space. Embedded hidden Markov model (E-HMM) was used by Zhong et al. [24] for synthesizing sketch out of photo. A similar method of synthesis was presented by Gao et al who used embedded model of hidden Markov in their work of research [25]. Method of E-HMM synthesis was improved by Xiao et al. with breaking of images in patches and focusing upon conversation of sketch to photo rather than photo to sketch. A method of synthesis or hallucinating faces was presented by Wei et al that builds upon non-linear method. Intrapersonal variation which occurs in sketches drawn by various artists was presented from Al Nizami et al. [26].

Klare and Jain [27] presented feature-based method to match sketches. In feature based approach of sketch matching both photo image and sketch are uniformly sampled using SIFT features descriptors at various scales [14]. Two separate featured-based methods were presented by Klair and Jain to match sketch with corresponding photo. These were common representation matching and direct matching. In the method of direct matching distance present between SIFT descriptors in photo domains and the sketch was measured. It was revealed that SIFT descriptors are greatly invariant towards photo and sketch modalities allowing for positive matching through direct comparison of these descriptors. In common representation method training is carried out using training set of sketch-photo correspondences for first measuring distance of gallery photos and probe sketches with training set by use of SIFT representation features. Distance between photo and sketch is measured by determining their similarity with subjects in set of training. Using method of common representation removed needs of

directly comparing image descriptors with modalities based upon thinking that in case computed image features out of sketches of two distinct people are dissimilar or similar, then same features computed upon corresponding photos of same two persons will also be dissimilar or similar.

Face recognition systems are very important for law enforcement. It is highly efficient in terms of automatic identification of suspects and speeding up of transactions. Generally, the images obtained from color camera sensors are similar to those displayed in the data log. It is relatively easy to solve such problems. In some cases, images of ordinary suspects obtained from RGB camera sensors cannot be obtained. In such cases, sketch drawing is used. Sketch drawings of the same person are quite different when drawn by different people. For this reason, if sketch drawings are converted to RGB images, identification of the person becomes easier.

In this study, a residual CNN structure is proposed to obtain RGB face images using sketch drawing. CNN structures have a very suitable architecture for image processing problems and are used in many studies in the literature. This architecture, which is famous after [45] Krizhevsky and his colleagues won the Imagenet competition using the CNN structure [45], is used almost all image and video problems thanks to the convolution layer's suitability for image processing problems. CNN architecture automatically learns features from the image. Compared to hand-crafted features, automatic feature learning is quite successful. Because hand-crafted features require experience, expertise and knowledge. In addition, the specified features may not always represent the data successfully. For this reason, algorithms that can learn automatic features come to the forefront. Many studies on face recognition and sketch identification problems have used hand-crafted features. Tang et al. [46] used geometric features and eigenface vectors for face sketch recognition. Liu et al. [47] applied a mapping study according to geometric similarities between face photograph and sketch. Zhang et al. [48] proposed a coupled information-theoretic encoding based feature extraction method for automatic face-sketch recognition. Galoogahi and Sim [49] propose a new face descriptor based on gradient orientation for automatic face sketch recognition. In applications which is performed with hand-crafted features, images are classified with a hand-selected classifier. But deep learning automatically classifies the features it acquires on its own. Mittal et al. [50] use transfer learning with deep learning representation for sketch recognition. Zhu et al. [51] propose a new deep

learning framework that can recover the canonical view. Tang and Wang [52] have compared face sketch images with real images by suggesting an automatic image retrieval system in the CUHK data set. Bansode and Sinha [53] proposed a genetic algorithm based image generation system from color image to face sketch image. Pramanik and Bhattacharjee [54] used principle component analyze to extract feature from color images. Then, some classifiers, which are KNN and SVM, applied to match face sketch images in CUHK database.

Zhang et al. proposed generative adversarial neural network model for original to sketch image conversion. They were used 61 images for training and 27 images for testing in CUHK database. Their proposed method reaches 100% accuracy in testing images [65].

2.4 Matching Forensics Sketches with Mugshots

In comparison to photo-based facial recognition, there is limited focus upon sketch recognition & much of published research has been directed upon sketches drawn by hand. Initial research within the field concentrated upon matching the sketches which artist draws while looking at person himself or at corresponding photograph of the person (called as the viewed sketches). The studies over viewed sketches are divided in two categories: model-insensitive feature representation and modal transformation. Approaches in the modal transformation [18] convert images out of one modality (sketch) into another distinct modality (photo). Modal transformation methods include local linear embedding (LLE), eigen transformation, multiscale Markov Random Fields model and embedded hidden Markov model (E-HMM) [25]. These approaches have an advantage that traditional algorithms of face matching that are designed for purpose of target modality can be used following modal transformation. However, synthesized photo can only be recognized as pseudo-photo because of inferred content which it has. These methods of synthesis in fact, are mostly solving a problem that is more problematic than task of recognition.

Second approach of sketch recognition tries of learning or designing feature representations which lowers difference of intra-class caused because of modality gap whilst conserving interclass separability. The methods of representation that are involved in such category are mutual discriminant space, coupled spectral regression, coupled information-theoretic project & partial least squares [28-29].

It was demonstrated by Klare et al. [13] that modernized facial matches are able to easily match (with more than 95 percent accuracy) the viewed sketches (which artists draw while viewing photograph or person) with actual photographs. Klare et al. therefore directed their focus upon more realistic and challenging issue of comparing forensic sketches (which forensic artists draws from just verbal description) with actual photographs. Though both forensic and viewed sketches are made by artists, difference is present because the forensic sketches are made depending upon verbal description provided by the victim or eyewitness, instead of directly looking at photograph or person. During forensic sketch drawing, the witness is mostly not able to recall exact facial look of suspect. In addition, many times difference is present in understanding and depiction of the facial features between an eyewitness and the artist. This gives rise to additional challenges in matching forensics with face photographs.

The two major difficulties of facial recognition technology for forensic applications are 1) face retrieval by use of various expressions and poses 2) matching of forensic sketches with databases of face photographs. For apprehending criminals in case there is no available photo of suspect, it is necessary to provide solution for these difficulties to remove duplicates in various databases of government, including driving license, passport photos. The first problem of expression and pose can be solved through Principal Component Analysis (PCA), Wavelet Transform and algorithms of Linear Discriminant Analysis (LDA). Another problem is considered in which improvements & analysis are provided to match forensic sketches with huge galleries of mug shot.

2.5 Image Retrieval of Face Image on the basis of Probe Sketch with Sift Feature Descriptors

The solution of problems exists somewhere between image retrieval and face recognition as sketches which are drawn by pencil has completely different modality compared to face photos. Two related & interesting problems are evident here: similar extraction of visual features out of photos and sketches, and comparison features with pairs of sketch-photo. Feature-based methods are preferred for comparing of facial images with face photographs. Descriptors here are measured at particular discrete points like nose, ears, eyes etc. Such features are suitable in matching of sketch photos as they define scattering of edges direction in face & their grouping with geometric

positioning of marked key points that has information common with both photos & sketches. This enables making comparison of only the features which are prominent. This includes two parts: Training & testing. Training begins with manual annotation of key-point upon training set of corresponding pairs of sketch-photo. But to do automatically Active Shape Model (ASM) can also be utilized for provided probe sketch at testing time. It is statistical model of object's shape in training image that deforms for fitting to object in new image. It notes natural variability that exists in class of shapes [30]. Model is constructed through learning of variability patterns of annotated points in database of training. This approach is able to match more than 100 pairs of sketch-photo within a minute with accuracy.

2.6 Matching Composite Sketches with Face Photographs: Component Base Approach

Difference from sketches which artist draw using his own hands, composite sketches are made in one of numerous composite facial applications present at the law enforcement departments. For measuring similarity that exists between mugshot photograph and composite sketch component base representation is proffered. This socially is able to automatically recognize facial landmarks within composite sketches and face photographs with the help of active shape model (ASM). Then features are drawn for every facial component by the use of multiscale local binary patterns (MLBPs) and per component similarity is determined [31]. Finally score of similarity obtained out of individual face components, fused with yielding similarity score between face photo and composite sketch. Performance of matching is improved further through filtering large gallery of mug short images using information about gender [32].

This method greatly outperforms the COTS face matching and is applied for enforcement of law. Analysing variations that exist in composite sketch quality produced by two users with different cultural and ethnic background presents a suggestion that users who operate software for facial composites should be trained for reducing cross-race biasness in order to generate high quality sketches for potential suspect of different races.

2.7 Matching Face and Retrieval for Forensics Applications

There are various approaches for forensics face recognition and many challenges are present in improving results of matching and retrieval and for processing of images which are low in quality. Forensic facial recognition differs from normal portrait facial recognition. System of face recognition normally is designed for outputting degree of similarity that is present in two images of face. It also includes finding of key face landmarks such as eyes centre for normalizing & alignment of appearance of face. By use of landmarks primary features of face are extracted and excluded from subsequent detection process of facial marks. Face image is mapped with mean shape for simplifying succeeding process. Although demonstrated performance of this approach is not vigorous, but face marks provide better descriptive representation for accuracy of facial recognition in comparison to values that are acquired from traditional systems of face recognition. Forensic facial recognition most of the time needs stage of pre-processing for enhancement of image or specific matcher for performing recognition. As surveillance applications are increasing, the world of forensics is changing and progress in the field of facial recognition is helping to lead forwards. New paradigm has emerged for identification of suspect with the help of forensic sketches. These sketches can be transformed into digital image & matched automatically with mug shots and additional images present in data base- such as driving license photos for helping with identification. This automatic approach enabled thanks to progress of computer vision and algorithms of machine learning offer valuable content to the authorities which seek to quickly and accurately apprehend dangerous law breaking suspects. Law enforcement's requirement for system that is able to match sketches with photos has resulted in continued researcher over increasing accuracy of face recognition system.

Excessive research is being carried out within vision community for making face detection work for real time application and complementary work is that of viola and Jones [33]. Face detector of Viola and Jones has become standard in building comprehensive face detection system for real time. One limitation of Viola and Jones however is that it creates high rates of false positive (meaning that a face is detected even when it is not present) & false negative (not detecting face when it is present) when directly applied upon input image. For dealing with such problems, different

type of enhancements have been put forwards like use of skin colour filters (either post filtering or pre-filtering) in order to give additional information in the images that are coloured. Whilst many experiments have demonstrated feasibility to combine SCF and VJFD for reducing false positives, the methods still suffer high rates of false negatives since some regions of face are not focused by detector when it is directly applied on input image which has complex backgrounds [34-35]. This greatly reduces accuracy of face applications because of the fact that particular regions of face are missed and the stages of system cannot recover missed region of the face. Therefore, eventual failure or success of succeeding stages is dependent upon false negative.

Various researchers have made contributions to overcome the problems in Viola and Jones framework. In [36] interesting methods were proposed by the authors for reducing false detection through use of skin colour as stage of pre-filtering before applying Viola and Jones face detection. In [37] a hybrid method was proposed by researchers for reducing false positive in Viola and Jones face detector through use of skin colour face method of post-filtering in Hue, Saturation and Value colour space. For reducing lightning effects, in [38] an algorithm of illumination compensation was applied by researchers and Viola and Jones face detector and skin colour was combined for detecting face. A method of reducing rate of false positive was proposed in [35] by authors to keep high detection rate of Viola and Jones face detector in real applications through use of post-filtration method based upon color-based filtering of skin in RGB colour space. A particular attention has been paid on visual attention mechanism for quickly finding attractive regions of face & overcoming issue of false negative when both SC & VJFD are directly applied to frames of video which have complex backgrounds.

Computational models have been discussed in advanced robotics and computer vision for predicting significant areas in visual field [39-40]. A renewed interest has been observed by researchers in usage of visual attention for rapidly detecting small objects of interests in visual environment and it is suggested that Human face & texture mainly attracts attention independent of tasks [41]. One of the major attempts of simulating visual mechanism upon computer is represented by saliency-based visual attention. Several approaches have been put forward to give computational foundation upon idea of visual saliency that can broadly be classified in a) bottom up, b) top-down & c) hybrid or Bayesian model. In bottom-up method, local features are used for image

under consideration for finding the image location and these completely differ from neighbours [41].

Several hybrid models recently have been put forwards by several researchers. Such models try formulating human attention in framework of Bayesian, combining model of bottom-up & top-down contextual information of object appearance & location [42-43]. Mostly, these models, explicitly or implicitly, approximate probability density of responses of filter obtained out of local features (bottom-up) of given image and combine it with probability density of object shape & location developed out of training samples in Bayesian framework. Researchers in [40] proposed bottom-up visual attention model based upon three features of image: intensity, orientation & colour in different scales. Implementation of center-surround operations is done through difference in filter responses existing among two scale for obtaining set of feature maps for specific feature. Normalization of feature maps is done and they are combined linearly for constructing final saliency map. Bottom up model was proposed by authors in [42] based upon dissimilarity metric use. A graph upon the image was defined by them and they employed random walks for computing visual saliency. At spot saliency measure is proportional with frequency of visits at equilibrium of random walk. In [43] researchers proposed hybrid visual attention models that attempts of calculating measure which exploits natural images statistics. Difference of Gaussian filters (DoG) is used to calculate local saliency. Intensity of image is used as input and filter responses are estimated through multivariate generalized Gaussian distribution. However, such visual saliency models are normally too complex & require greater computational capacity [39]. It is important for these models to give fast response for applications that operate at high levels like the robotics vision and analysis of visual content. To offer real-time visual attention model, Butko et al. [39] presented simple version of model of Zhang et al [43]. They evaluated saliency model in saccades control of camera in social robotics situations.

CHAPTER 3

MATERIAL AND METHOD

3.1 Artificial Neural Networks

Artificial Neural Networks (ANN) are computational models that are inspired by the biological brain structure with complex processing capabilities [55]. They consist of many artificial neurons connected in a similar way to the biological brain structure. It has layers of ANN neuron groups. The ANNs can be composed of a single layer or can be developed in ANN, which can be composed of many layers. Today's computers can easily solve many algorithmic problems. But it is very difficult for computers to solve behaviors such as sight, speech, hearing that are easy for people. ANN is laying the groundwork for making computers behave like humans. Gain experience by learning from the data and use the experience gained to solve a problem. The history of ANN dates back to the early 1940s. In 1943, models of threshold switches of neuron networks based on neurons were presented by Warren McCulloch and Walter Pitts. It has also been shown that even simple networking models can calculate almost any logic or arithmetic function [56].

3.1.1 Single Neuron Model

The neuron is the simplest unit of calculation. It takes an input value and generates an output value. As in Figure 3, the input values are multiplied by specific weight values, and if present, the neuron's bias value is added to obtain the total. Total number of all neurons in any layer are defined as follows:

$$T(x) = \sum_{i=1}^3 w_i x_i + b \quad (0.1)$$

$$T(x) = w_1 x_1 + w_2 x_2 + w_3 x_3 + b \quad (0.2)$$

The total value obtained is given as input to the activation function and is activated according to the neuron outcome value.

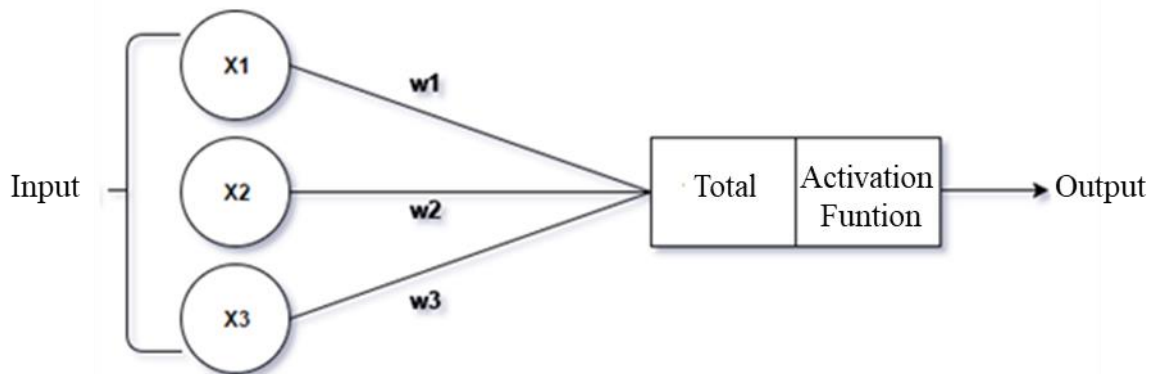


Figure 3 Calculation of neurons

In ANN, weight and bias values are parameters that can be learned. The bias values in each neuron are intended to enable the activation function used to move on the axis. Even if the weights influence the graph of the activation function, the bias values can be used to express the different states of the network.

3.1.2 Activation Functions

The activation functions are referred to as the Non-Linear Function in ANN. As its name implies, since the linear function is not generally chosen as the activation function, it gives the neurons a nonlinear structure and gives them the ability to represent more complex functions. If the use of linear functions is preferred in the choice of activation function, the network can express linearly solvable problems.

Neurons act according to activation functions. Neurons with values that fall below a certain threshold value according to the selected activation function are not activated. Inactive neurons are not used.

Various activation functions are available. From these functions, ReLU (Rectified Linear Unit) is widely used in modern deep models. Sigmoid and Tanh functions are also preferred as activation functions.

3.1.2.1 ReLU

The expansion of ReLU is the Improved Linear Unit. As defined in Eq. (3.3), thresholding is performed by assigning negative values given as input to the function

to zero and values greater than zero as their values. The most important feature that separates this function from others is that the activation process is performed quickly. The graph of the function is given in Figure 4.

$$\omega(x) = \max(0, x) \quad (0.3)$$

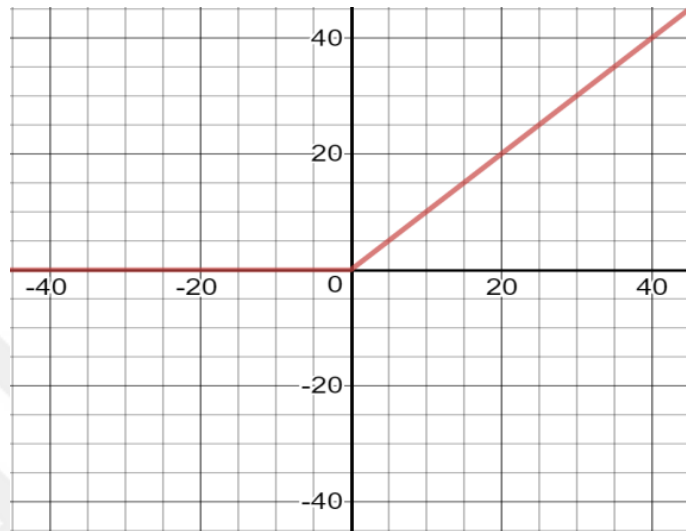


Figure 4 Graph of ReLU activation function

3.1.2.2 Sigmoid

The sigmoid function expresses the input values as 0 and 1. A graphical curve is obtained from the function as shown in Figure 5.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (0.4)$$

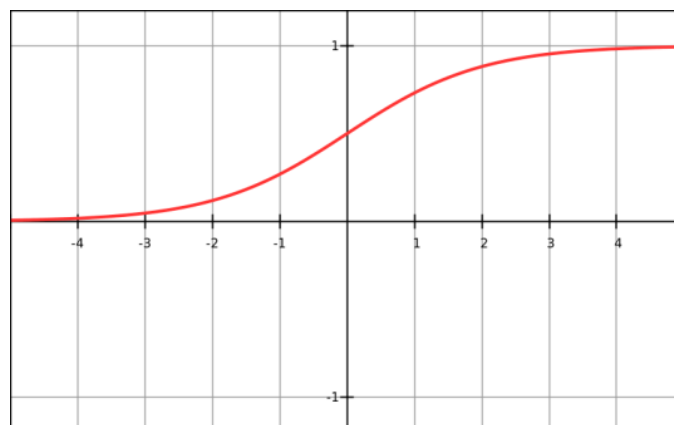


Figure 5 Graph of sigmoid activation function

3.1.2.3 Tanh

Tanh function resembles Sigmoid function. It expresses input values differently between -1 and 1.

$$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (0.5)$$

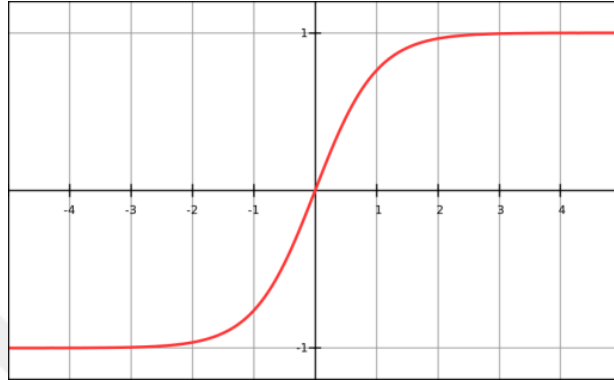


Figure 6 Graph of Tanh activation function

3.1.3 Feed Forward Networks

Feed Forward Networks are composed of layers containing different number of neuron groups and information flows to the advanced layers. The neurons in the network are called nodes. The nodes in the layers are linked to each other by links containing weight values. The nodes in the same layer are not linked to each other, as this can create a loop on the network and prevent forward transmission. There are three different types of layers, the input layer, the hidden layer and the output layer.

Data input is provided according to the number of nodes determined from the input layer. A hidden layer is a layer between the input and output layers. This layer is optional. The number of hidden layers can be increased especially when the network is desired to be deepened. The input values are calculated and forwarded on the network to provide forward information flow and values are transmitted to the output layer.

It is possible to explain directed acyclic graphs of how the functions in the forward feed network work together. A function $f(x)=f^{(3)}(f^{(2)}(f^{(1)}(x)))$ can be given when defining three functions connected in a chain. This chain structure is a common structure in ANN. When $f^{(1)}$ is defined as the first layer, it will refer to the other layers

in the other functions respectively. The total length of the chain gives depth of the model [34].

3.1.3.1 Single Layer Perceptrons

Single layer detectors are a feed-forward neural network type with no hidden layer. The input layer is connected to the output layer. They are used to solve uncomplicated linear problems because they can represent linear functions. Figure 7 shows a single layer perceptron model.

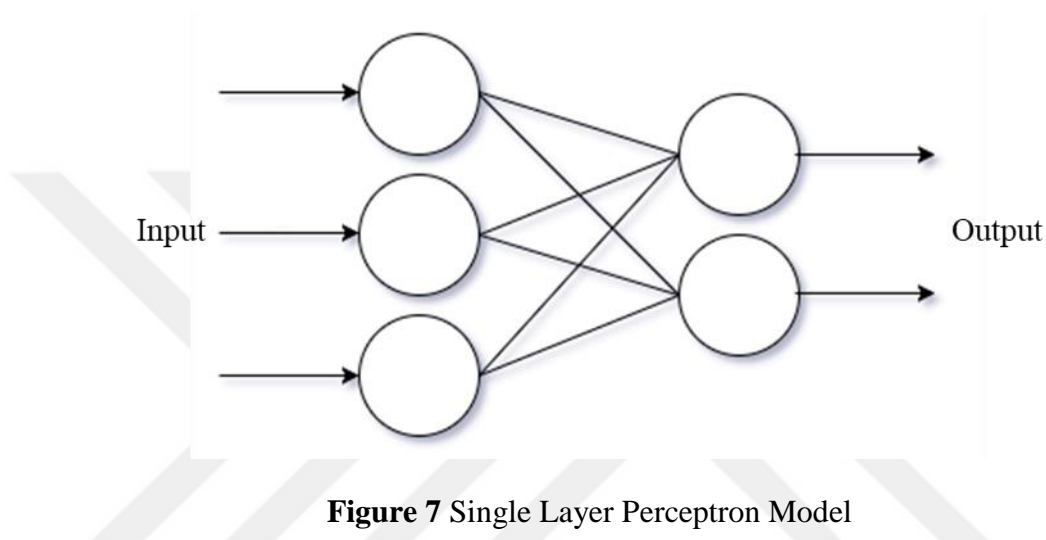


Figure 7 Single Layer Perceptron Model

3.1.3.2 Multi-Layer Perceptrons

Multilayer perceptrons are a type of feedforward neural network with at least one hidden layer. Hidden layers ensure that the properties in the data are determined. The more neurons in the hidden layer are, the more features can be deduced. Non-linear functions can be represented by hidden layers. Multilayer detectors are used to solve problems that are more complex than the problems that single layer detectors can solve. Figure 8 shows a multilayer sensor model with one hidden layer.

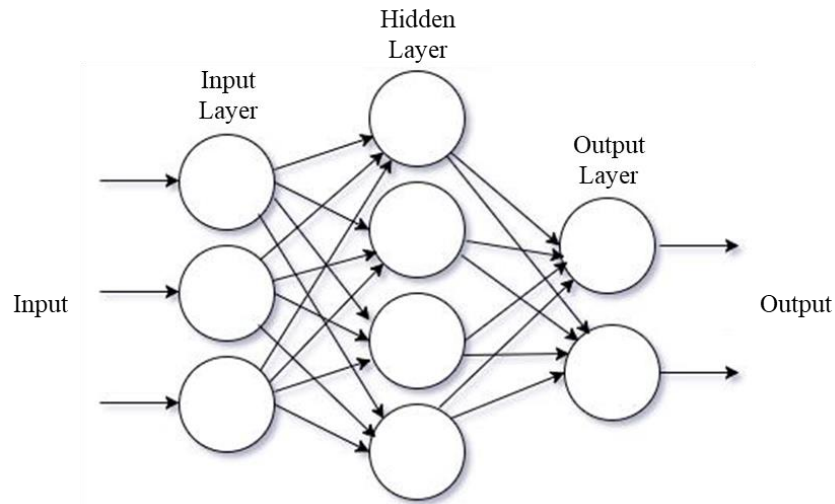


Figure 8 Multi-Layer Perceptrons Model

Using the sigmoid activation function, the calculation of a neuron in a hidden layer in a forward feed network as in Figure 8 is performed as follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (0.6)$$

$$T_2(x) = \sum_{j=1}^{N=3} (w_j x_j + b) \quad (0.7)$$

$$O_2(x) = \frac{1}{1 + \exp(-(w_1 x_1 + w_2 x_2 + w_3 x_3 + b))} \quad (0.8)$$

The sigmoid function $\sigma(x)$ is given as input to the sigmoid function $T_2(x)$ which indicates the total function of the data from the three neurons in the input layer. The resulting $O_2(x)$ value represents the output value of neuron number 2. When this value is zero, the neuron will be active and will be activated and ignited.

3.1.4 Back Forward Networks

Backward ANN does not consist of a structure in which all the neurons in the layers, such as feed forward ANN, are sequentially connected to the neurons in the next layer. The output of a neuron can be the input of a neuron in the previous layer. Feedback networks are referred to as repeat networks. They are often used to solve sequential problems such as time series.

Feedback networks can carry signals in both directions by creating circles in the network. These networks are quite powerful and can be extremely complex. Dynamic states and conditions are constantly changing until reaching the equilibrium point. They remain in equilibrium until the input changes [57].

The simple repetitive network model called the Elman network was first designed and used by Jeffrey L. Elman in 1990. It has a network structure that is formed by adding a context layer to the structure of the forward feed ANN. The activations are copied constantly from the hidden layer to the context layer, with the weights being constant. Hidden units encode the patterns that allow the network to produce the correct output for a given input and generate internal representations for the input patterns. Thanks to the context units, the previous internal situation is remembered. The units in the hidden layer have the task of mapping both an external input and the previous state of some desired output. The patterns on the hidden units are recorded as context. Thus, the inner representations become sensitive to temporal context. This model naturally involves the processing of input submitted in turn, respectively [58, 59]. Such models can be used in applications that can do weather forecasting.

3.2 Deep Learning

Deep learning method is a popular machine learning approach which is evolved by the deepening of ANN methods. In this method, the calculation structure is based on ANN calculations. It is a better method than classical ANN approaches. The number of layers in the models was kept to a small number because it was difficult to train ANN because hardware facilities were limited in previous years. It became possible to train deeper models with the introduction of machines with high processing power. Especially, models have been started to be trained on GPUs and the deeper models have been trained in a shorter time than the CPU because of the GPU 's ability to make matrix operations fast.

One of the most important factors in the popularization of deep learning is the increasingly large quantity of processor abilities and the increase in the number of data used for training [60]. For example, On the Facebook platform, approximately 350 million images are uploaded each day, while 100 hours of video is uploaded per minute on Youtube. There is no doubt that the importance of data that reflect the real world is

invaluable in enabling computers to solve problems better because they have gained experience from deep learning [61]. Increasing data on the Internet, such as images, video, audio, text, etc., have reached the dimensions that reflect the real world. With the use of a large number of different data, more effective training has become possible.

The problem solving ability of deep learning comes from the ability of ANN to solve non-linear problems. The number of layers increasing parallel to the depth of the model provides the basis for solving complex problems. Deep learning models also provide flexibility in solving problems in different areas with few changes. Speech recognition, natural language processing as well as deep learning are frequently used in computer vision applications for classification, segmentation and feature extraction in recent years.

3.2.1 Image Classification with Deep Learning

The computer vision perceives the external environment as pixel matrices at certain dimensions. Pixel matrices consist of the values determined by the number of bits of the image. When classifying complex views of human vision, it is necessary for computers to perform this operation in the middle of a second with matrices, which requires both sufficient power processing and an algorithm to optimize this operation.

Pixel matrices of an object in the image can change depending on many factors such as angle of arrival of the light, the objects that can be perceived as three-dimensional in the outer world, the two-dimensional appearance as a different object from different angles, and the closure of a part of objects by another object. For this, every feature of the object in the image needs to be analyzed in a good way. When using a verbally learning approach it is necessary to determine which properties belong to which data. In addition, the specified properties must be distinctive features that differ from the other objects.

One of the most important features of deep learning is that, unlike classical machine learning techniques, there is no need for feature engineering work. Upon deep learning, feature extraction is automatically performed on the data. Convolutional neural networks are widely used in image classification problems. In these networks, the image is analyzed by matching to find the features in the image, and while the trained

network analyzes a complex scene, it can detect a limited part of the object, but it can perceive to which object the relevant part belongs through the property mapping.

3.2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNN) is basically composed of two parts. The first section is where the feature extraction is made up of one or more convolutions and pooling layers. The second part is the part of the classification process which consists of fully connected layers. Figure 9 shows CNN stages.



Figure 9 CNN stages

These networks are mostly used in image classification field. Predictions are carried out by making certain property deductions such as the human vision mechanism in the network. It is a feature extraction from image pixels which are difficult in image classification problems. This process, which is difficult for CNN, automatically runs in the first section to obtain feature maps. Then, the acquired properties are transmitted to the fully bounded layers, and by means of the multi-layered sensors, certain numerical values are obtained and relevant estimates are made about which class belongs to the class.

In computer vision applications, the pixel values of the image at the input layer of CNN are used as input. The number of neurons in the input layer varies depending on the number of color channels. A color image has three neurons of red, green, blue color channels in a color image while the input layer in a monochrome grayscale image has a single neuron.

When pixels of an image are taken as input, they are transformed into multidimensional matrices called tensors. A $1 \times 3 \times 200 \times 200$ tensor can be used for a color image with width and height values of 200. The reason for this is to make the network perform certain matrix operations.

3.2.2.1 Convolution Layer

The purpose of convolution layers is to extract property maps using pixel matrixes of the view. In this layer, the filters are applied by shifting the filters (weights) in different dimensions, such as 3x3, 5x5, starting from the upper left and lower right corner of the pixel matrix of the view. When filters are applied, each scrolling operation is multiplied by the selected pixel matrix at the same size as the filter, and the resulting matrix elements are summed to obtain the result. The feature maps are created after the process is repeated until the last pixel arrival.

The number of filters applied from the input layer to the next layer is the depth of the view, and provides low level feature extraction. In later convolute layers, high-level features including specific parts of the image are determined. Determining the average size of the features to focus on the image can be used as a factor in determining the filter size to be applied in the first layer. Filters provide multiple versions with a focus on various aspects of a scene. Applied filters consist of weights. Weights are learned by network training, the part of the feature extraction is CNN, which differs from other artificial neural networks. The more convoluted layers are found in the network, the more complex characteristics can be determined.

For example, assume that a three-channel 50x50 image is applied to the first convolute layer with 5x5 size filters. Once each channel is filtered, feature matrices of size 46x46 will be obtained. When scrolling is applied, stride and padding parameters are used in addition to the filter size. The step parameter specifies how long the filter will be slid while filtering. The space parameter is how the edge values are processed.

3.2.2.2 Pooling Layer

The pooling process is implemented at the later stage of the filtration process in the convolution layer during the property extraction phase. The purpose of this process is to reduce the size of the resulting feature maps. It is also used to avoid overfitting problems.

As in the convolution layer in the pooling process, the calculations are made on the matrices using filters in various dimensions. By choosing the value of the double frame, taking into account the determined step parameter, two different operations can be performed, namely maximum and average pooling on feature matrices. Maximum

pooling layers were used in this study. In the maximum pooling layer, the largest value of matching values in the pooling frame and the feature matrix is taken. In the average pooling layer, the average of the matching values in the pooling frame and feature matrix is taken.

3.2.2.3 Classification Layer

The categorization phase takes place after CNN's feature extraction phase. This phase takes feature maps as input. The fully bonded layer providing the classification is designed in a multilayer sensor construction. Since the values to be obtained at the exit of the network are numerical values, after processing with multidimensional data (matrices), the dimension is reduced in this layer to make the data one-dimensional.

Binary classification and multi-class classification can be performed in two ways. There are two classes in the binary classification. When an object is found in an image, positive marking can be detected, and in the absence of an object, negative marking can be performed to determine whether the object is in the image. In multiple classification, there are n classes and the aim is to determine which one of the n classes belongs to a given data. An example is an ANN binary classification that estimates that the car is a sport or a regular car based on the passenger capacity, the number of doors and the height of the car. An example of an ANN that carries estimates from n flower classes by looking at the petal leaf, leaf, and color characteristics for multiple classification.

In the problem of classification, the net output is equal to the number of classes used in education. In the output layer, probability values can be used that indicate which class is more associated with which class to display from the feature map. The softmax function is used to obtain the probability values.

Softmax function is used in the last layer of CNN. The reason is that the numerical values obtained are related to which classes.

$$f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \quad (0.9)$$

The function takes a number sequence as input and produces output between 0 and 1 as output. Since the probability expresses, the sum of the result values must be 1. In

multi-class classification, the class with the greatest value as a result is regarded as the correct guess. In this study, because of multi-class classification, Softmax function was used as the activation function in the last layer.

3.2.3 Learning Process

As in the techniques of machine learning, learning from the data is also a matter of learning deeply. Learning processes are divided into supervised, uncontrolled and reinforced learning. In all processes, data forms one of the most important parts of the system. The problem to be solved is that the problem must be diverse to reflect the true world, depending on the type. When training is done, the weight values are updated to give the desired results and the most appropriate mathematical function is obtained on the network that can distinguish the data according to a certain level. A model developed and trained to solve any problem can be used to solve a different problem when re-trained with different data.

3.2.3.1 Supervised Learning

Data for supervised learning is given in a labeled format to the machine. That is to say what will happen depends on the specific input values. Supervised learning is also used in classification and regression problems. The network model is trained by supervised learning, and learning is carried out with a large number of data that contain different conditions. The purpose of learning is to ensure that the error rate of the system is reduced. While the error rate is reduced, the net weight values are updated to provide the desired conditions. Estimates are made with minimum errors according to specific inputs. The error value is expressed as the difference between the expected value and the actual value.

3.2.3.2 Unsupervised Learning

The data given to the machine during the unsupervised learning process are untagged. It is not told which output values should be obtained, as opposed to input values, as in supervised learning. Instead, the system learns by giving meaning to data itself. When the network is updated, the weights are also updated according to the network rules. This method is used to determine patterns from a large number of data.

The main idea of learning deeply is to learn not only nonlinear mapping between inputs and outputs, but also what lies behind input vectors [62]. ANN uses autoencoder for learning without supervision. The autocoders do not learn the data with the labels and give information about the data that they receive as input. When this method is used on image data, there are 2500 neurons inputting all the pixels in the input layer for a 50x50 grayscale matrix. All pixel values are used to determine the property by hidden layers.

3.2.3.3 Reinforced Learning

The labels of the data are not known as in the case of learning without supervision in the reinforced learning process. However, the system is helped to improve itself by saying that the resulting output is true or false. If a correct result is obtained after a certain number of steps, the penalty signal is sent when a wrong result is obtained. An example of a reinforced learning process is when a child touches a hot object such as a stove and learns that the nervous system feels painful to feel a painful object and touches the object when the same scenario is repeated. Computer intelligence, which can usually play games when learning deeply, is used to create game agents.

3.2.4 Selection of Data Sets

In the selection of data sets, the data are basically divided into training and test data. The generated model is trained with the training data, whereas the success of the model is evaluated with the test data that have never been seen before. When training is conducted in the supervised learning process, the value of the cost function is minimized and the function to represent the data in the best way is tried to be obtained according to the education data. The trained model sometimes only represents the training data set. In this case, testing with new data is less predictive. Since the training data is memorized, the model can only interpret the examples included in the training data set as success.

It is undesirable that the accuracy obtained from the test data sets is much lower than the accuracy obtained from the training. This is called overfitting. In some cases, the trained model also makes unsuccessful estimates in the training set. This probing is called underfitting. Desirably, the model provides results that are close to both training and test data, so that it is possible to predict how well the data can be predicted.

Sufficient variety of data can be used in sufficient numbers to avoid excessive adaptation problems to deep learning. The complex model can be simplified, or the model can be prevented from memorizing the training data with too many repetitions. In order to do this, the training can be stopped when the sufficient number of repetitions is reached, which is called early stopping. Regularization techniques are also used. In this study, dropout technique is used as regularization technique to avoid these problems.

Data sets can be categorized as training, test and verification data sets. While the model is being trained, the over-compliance problem can be detected using the validation data set. The verification dataset is used to adjust the parameters to improve the model by understanding how the designed model gives results on the data outside the training data. Therefore, the use of the verification data set is important to achieve real performance.

3.2.5 Cost Functions

Cost functions are also expressed as error functions. These functions represent network failure and are minimized during the training process. There are various cost functions. Cost function selection is based on probing. In regression problems, mean square error, mean absolute error, and root mean square are used, while cross entropy cost function is used in classification problems.

3.2.5.1 Mean Squared Error

Mean squared error is widely used in regression problems.

$$MSE (w) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (0.10)$$

$$MSE (w) = \frac{1}{N} \sum_{i=1}^N (y_i - f(x_i, w))^2 \quad (0.11)$$

$$MSE (w) = \frac{1}{N} \sum_{i=1}^N e_i^2 \quad (0.12)$$

Using existing training data, the weights of the network can be used as a performance criterion. From the training data, for each instance i th, x input is taken together with weight values, and the value of the output value of the network is obtained. The expected output value y of the network is compared with the value \hat{y} and e is found by calculating the difference. To find the average error value, the square of e is taken, the number of training data is summed up and averaged.

3.2.5.2 Mean Absolute Error

When mean absolute error is calculated, the absolute value of each error is taken as the number of data and the average is calculated. The resulting value is obtained as follows.

$$MAE(w) = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (0.13)$$

$$MAE(w) = \frac{1}{N} \sum_{i=1}^N |e_i| \quad (0.14)$$

In this method, it is aimed to approximate the value of output \hat{y} to the value of y as if the mean error rate is present.

3.2.5.3 Root Mean Square

When mean error rate is calculated, the error rate is multiplied by the number of data, and the result is summed and the result is the square root. It can be expressed as follows.

$$RMS(w) = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (0.15)$$

$$RMS(w) = \sqrt{\frac{1}{N} \sum_{i=1}^N (e_i)^2} \quad (0.16)$$

Square root mean and mean absolute error are frequently used interchangeably. The root mean square helps to fully describe the error distribution of the entire network. It is more convenient to use with large error values when there are at least one hundred and more samples at our disposal.

3.2.5.4 Cross Entropy

The cross entropy function is a cost function used in classification problems. It is defined as follows.

$$CE = -\frac{1}{N} \sum_{i=1}^N (y_i \ln \hat{y}_i + (1 - \hat{y}_i) \ln(1 - \hat{y}_i)) \quad (0.17)$$

Thanks to the negative sign at the beginning, it does not give negative results like the cost functions explained before. ($CE > 0$) When a neuron output \hat{y} is close to the expected value y , the CE result will be close to zero. Unlike other cost functions, it provides a solution to the problem of learning slowdown.

3.2.6 Gradient Descent

Gradient Descent is a method of improving that allows the network to learn in ANN and deep learning methods. It provides the weight and bias values of the network in order to reduce the cost function to a minimum. Equation 3.18 gives the weight update function.

$$w_{t+1} = w_t - \eta \nabla C(w_t) \quad (0.18)$$

Weight and bias values are updated after each epoch. Figure 10 shows a change in the value of a cost function depending on a single weight. The point indicated by the first point i of the weight value. Starting from this point, the weight is gradually updated and the lowest cost function value is tried to be obtained. When the weight is updated, the multiplication is performed by the educational stake, that is, the weight of the cost function, with the derivatives $\nabla C(\mathbf{w}_t)$ and η learning rate. The obtained value is subtracted from the weight value \mathbf{w}_t and the new weight value \mathbf{w}_{t+1} is obtained. The learning rate is used to determine how large the weights are to be updated during

training, thus determining the size of each step. When a low value learning rate is determined, it will take time to reach the saturation point according to the good learning rate. When a large value learning rate is determined, updating is done in the direction of the visual arrows shown on the right in Figure 10. The leaping movements to the right and to the left move away from saturation point. Achieving the saturation point with an appropriate learning rate is as seen on the left.

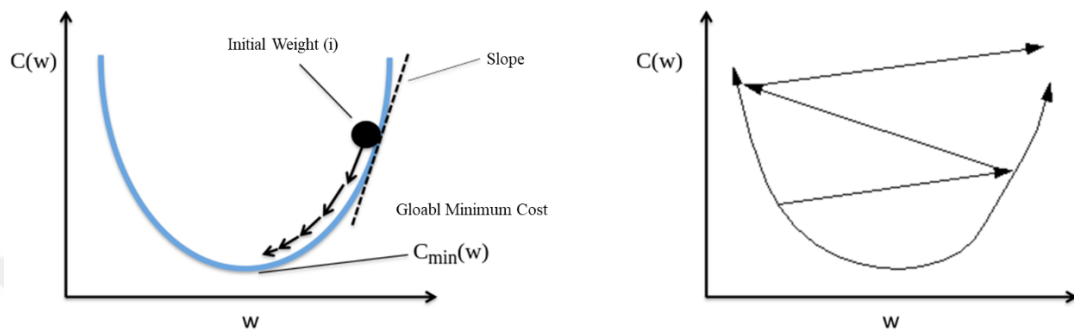


Figure 10 Access to saturation point

3.2.7 Momentum

In practice, it is difficult to find the smallest spot quickly, with only a few steps, by setting only one weight parameter. Saturation can be achieved by doing a lot of repetition with the setting of millions of parameters for this. In the case of Figure 11, there is a specific area, which has the lowest points in itself. The lowest point is the local minimum name. When making slope reduction algorithm updates, steps are taken towards the point lower than the previous point. If the randomly initiated weight value is started near the local minimum, the local minimum will be selected as the lowest point. The point to be reached is not the minimum local points but the smallest point compared to all minimum points. This point is called the universal minimum. In complicated situations, it is not enough to simply descend to reach this point.

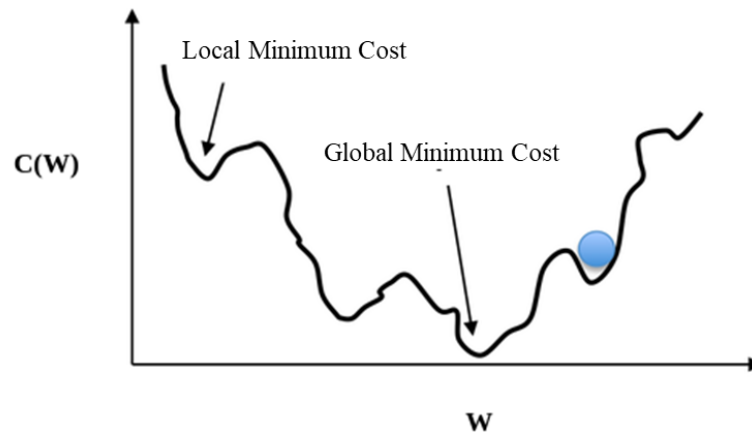


Figure 11 Local and Global Minimum

There is a technique called momentum that helps to reach the universal minimum by overcoming local minimums. As in Equation 3.19, the application of the momentum value, which can be between 0 and 1, by multiplying the weight value by the multiplication of the weight value makes the updating process quicker by taking bigger steps to reach the smallest spot without attaching the local point, and thus the learning period is shortened.

$$w_{t+1} = aw_t - \eta \nabla C(w_t) \quad (0.19)$$

When the momentum value is small, the learning period is long and the desired point can not be reached and the local minimum point can not be reached. When the value is large, saturation point can be passed. There is an inverse relationship between the learning rate and the momentum, and it is more appropriate to use a large learning moment and a large momentum.

3.2.8 Stochastic Gradient Descent

The stochastic gradient descent is a different version of the gradient descent algorithm. In the slope reduction algorithm, the slope calculation of the cost function is performed with all training data and the updating of the parameters is performed by taking the average. In the case of the statistical algorithm, one or more random samples are selected from the training data instead of performing the operation on the entire data

set and updated according to each example. Selected samples are processed until the end. In cases where large data sets are used, it is costly and time consuming to perform a one-time calculation for all data sets in the slope reduction algorithm. Performing these operations on specific examples speeds up the process. A weight update operation for n entries of x entries selected from the training set can be expressed as follows for each x in n instances.

$$w_{t+1} = w_t - \frac{\eta}{n} \sum_i \frac{\partial C_{x_i}}{\partial W_t} \quad (0.20)$$

3.2.9 Back propagation

In the ANN and deep learning techniques, back propagation algorithm is used besides the slope reduction algorithm in the network learning process. The backpropagation algorithm ensures that the derivatives are computed effectively. Partial derivatives of the weight function or bias value of the cost function are used when updating the weight and bias values in network training. The purpose of using the partial derivatives is to calculate the effect of the learned parameters in the system on the error value.

In networks with different numbers of hidden layers, each layer affects the error and each layer has its own fault. The error of the last layer is simply the value obtained by the cost function. The error of hidden layers is defined in a different way. To update the weights of the previous layer in back propagation, use the next layer's fault. The error is called back propagation because it propagates backwards starting from the output layer. Weights updates vary between output layers and hidden layers. Error variants are calculated for all hidden layers. When we get error derivatives for hidden layers, it is easier to calculate the derivative of weights based on hidden layers. Transactions are based on chain rule.

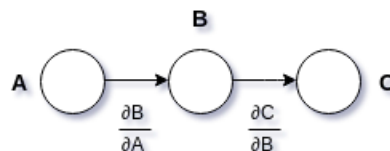


Figure 12 Simple chain rule

In a small neural network model with layers A, B and C, a simple chain rule expressing the back propagation of the error obtained from the C output layer can be defined as follows.

$$\nabla C = \frac{\partial C}{\partial B} \nabla B \quad (0.21)$$

$$\nabla B = \frac{\partial B}{\partial A} \nabla A \quad (0.22)$$

$$\nabla C = \frac{\partial C}{\partial B} \frac{\partial B}{\partial A} \nabla A \quad (0.23)$$

$$\frac{\partial C}{\partial A} = \frac{\partial C}{\partial B} \frac{\partial B}{\partial A} \quad (0.24)$$

In the backpropagation algorithm, all of the weight values connecting neurons in the layers are calculated. Below is a neural network model with a hidden layer containing three neurons. The m and n neurons were selected from the hidden layer and output layer of the model. When error accounts are made between these neurons, the paths of the weight values to be processed are indicated by straight lines, and the paths not to be processed by the broken lines.

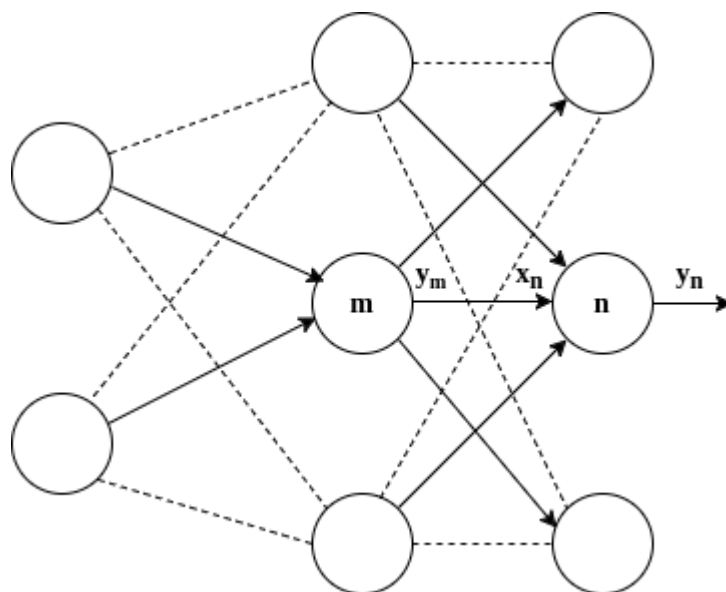


Figure 13 Back propagation on neural network

The input of n neurons in the last layer is the sum of the weight paths connected to the output of the neuron in the hidden layer and is expressed by x.

$$x_n = b_n + \sum_m y_m w_{mn} \quad (0.25)$$

Derivatives to be used in the propagation of the error can be defined as follows.

$$\frac{\partial x_n}{\partial w_{mn}} = y_m \quad (0.26)$$

$$\frac{\partial x_n}{\partial y_m} = w_{mn} \quad (0.27)$$

When we use sigmoid function in activation processes, the output of n neurons can be expressed as follows.

$$y_n = \frac{1}{1 + e^{-x_n}} \quad (0.28)$$

The partial derivative of the activation function according to x_n is as follows.

$$\frac{\partial y_n}{\partial x_n} = y_n(1 - y_n) \quad (0.29)$$

In the back propagation of the error, some derivatives can be defined as follows according to the chain rule.

$$\frac{\partial C}{\partial x_n} = \frac{\partial x_n}{\partial x_n} \frac{\partial C}{\partial y_n} = y_n(1 - y_n) \frac{\partial C}{\partial y_n} \quad (0.30)$$

$$\frac{\partial C}{\partial y_m} = \sum_n \frac{\partial x_n}{\partial y_m} \frac{\partial C}{\partial x_n} = \sum_n w_{mn} \frac{\partial C}{\partial x_n} \quad (0.31)$$

$$\frac{\partial C}{\partial w_{mn}} = \frac{\partial x_n}{\partial w_{mn}} \frac{\partial C}{\partial x_n} \quad (0.32)$$

3.2.10 Regularization Technics

Neural network models used in deep learning techniques can have a problem of excessive adaptation because they have millions of parameters or are not trained with sufficient data. Techniques used to solve this problem are called regularization techniques. Commonly, L1, L2 and Dropout regularization techniques are used.

3.2.10.1 L1 Technique

For the L1 regularization technique, weights are added to the cost function. Major weight values are penalized to zero. When the penalty term is obtained, the absolute value of each of all weights is taken, the average of these values is calculated, and the result is multiplied by the regularization constant (λ). The equality to be obtained by adding this term to the cost function is as follows.

$$C(w) = C(w) + \frac{\lambda}{n} \sum_{i=0}^n |w_i| \quad (0.33)$$

3.2.10.2 L2 Technique

As in the L1 technique, it is also aimed to reduce the weights by adding the term to the cost function in the L2 regularization technique. However, when the penalty value is calculated, the squares of each weight are taken, the average is calculated, the value is multiplied by the regularization constant (λ), and the result is divided by two. The cost function can be expressed as:

$$C(w) = C(w) + \frac{\lambda}{2n} \sum_{i=0}^n w_i^2 \quad (0.34)$$

The L2 regularization technique prevents the use of unnecessary weights in the network. Greater weight equivalence in the L1 technique can be achieved to better represent the data by the network.

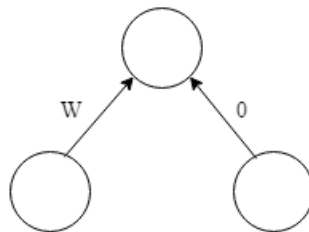


Figure 14 Weights Selection

In L1 and L2 regularization techniques, although the weight reduction process is performed in common, the shape of weight reduction is different. While the L2 technique reduces the weight in proportion to the weight (w), the weights in the L1 technique are reduced to zero constantly. When the absolute weight ($|w|$) is large, L1 performs less weight reduction than the L2 technique, and when the value is small, L1 can do more weight reduction than L2. In the L1 technique, when weights with high prefixes are changed little, it is desirable to reduce the other unimportant weights to zero.

3.2.10.3 Dropout

Dropout is the most commonly used regularization method in deep learning models. Unlike the L1 and L2 techniques, some of the neurons on the network are randomly deleted while training is being performed. In this way, neurons are prevented from representing the same features, allowing each neuron to represent different characteristics, enabling the network to operate more effectively.

Combining several models in machine learning techniques often improves performance. Many different models of deep learning techniques are costly to be trained separately and to be averaged. Different deep nets. it is difficult to find optimal parameters for each one and each requires more computational power than non-deep networks. The dropout technique enables efficient integration of many neural network models. The use of dropout regularization techniques in the training of neural networks developed for image and speech recognition, document classification, and computational biology problem solving has been observed to improve performance by providing good results on many datasets [62].

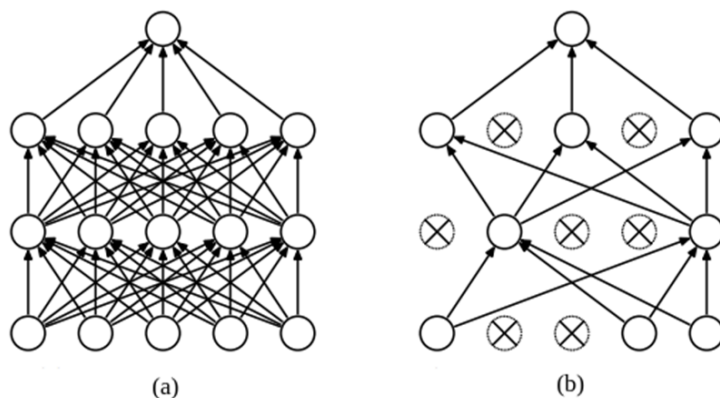


Figure 15 a) Conventional Neural Network b) Dropout Application

It allows the neural networks in one layer to be randomly deleted according to the selected probability value between 0 and 1. Figure 15 shows a network that has been regularized using a dropout technique with a probability value of 0.5. When dropout is used, it takes longer to reach the saturation point of the model. This technique is only performed when training. During the test, deletion is not performed. The output weights of the neurons that were deleted during training are multiplied by the probability value and advanced calculation is made.

3.3 Data Description

The CUHK Face Sketch Database (CUFS) is used to train and test the proposed system. This set includes 188 sketch and face images. Of these images, 148 were randomly selected for training. Then 148 image data augmentation operations are applied. On this count, the training data is multiplied. Image resolutions are 200x250 pixels. All images have 3 channels (Red, Green, and Blue). Some images in the data set used are shown in Figure 16 which depicts the original images in the dataset and the images obtained by data augmentation.

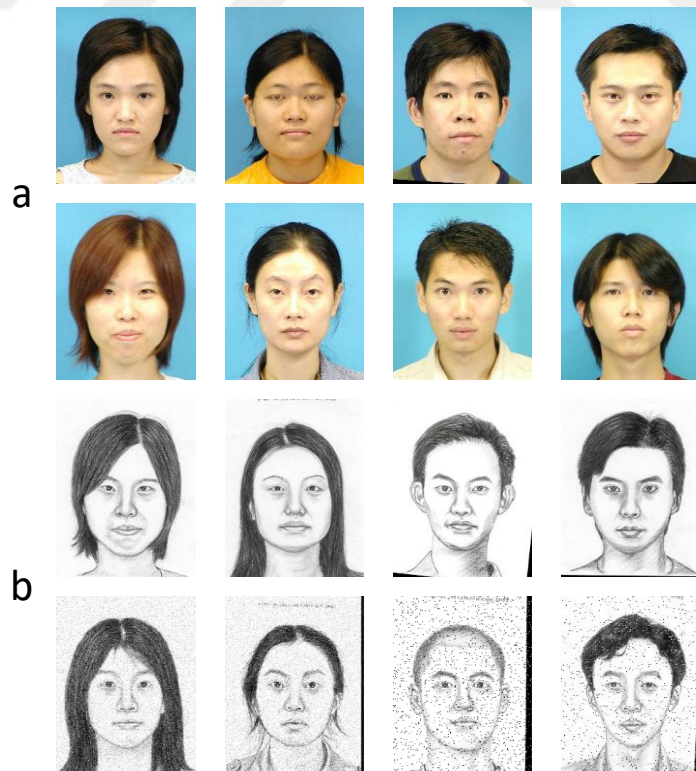


Figure 16 Image Dataset. a) face images b) sketch images

Also, AR dataset used for testing process with 123 images. Their image resolutions are 100x150 pixels. These have 3 colors channel as Red, Blue and Green.



CHAPTER 4

PROPOSED METHOD

Face detection is a quite complicated process due to many challenging reasons such as facial look variations, expression and illumination changes. Sketch matching is considered as a new challenge on top of all these challenges. In the face sketch process, the sketch is made according to the descriptions given by a witness. In this case, all facial expressions depend on personal interpretation. The sketch-face recognition process, which is crucial for criminal investigation, can even vary depending on the psychological state of the examiner. In this case, face recognition becomes extremely difficult. Another difficulty is the ability to distinguish sketch drawings that are automatically generated by computers. To distinguish fake images will always be challenging. All these difficulties are quite misleading and confusing for the human eye. It is possible to deceive artificial intelligence methods using mug shots. For this reason, it may be a solution to obtain possible face photographs from sketch images and compare them to the database. In the comparison process, a similarity confidence coefficient must be specified. This confidence coefficient can be used to avoid possible counterfeit drawings.

Sketch images are applied to the proposed network entrance. At the exit of the network, RB face images are obtained. The main problem here is that the obtained facial photograph can be matched with the images on the dataset. The problem mentioned is shown in Figure 17.

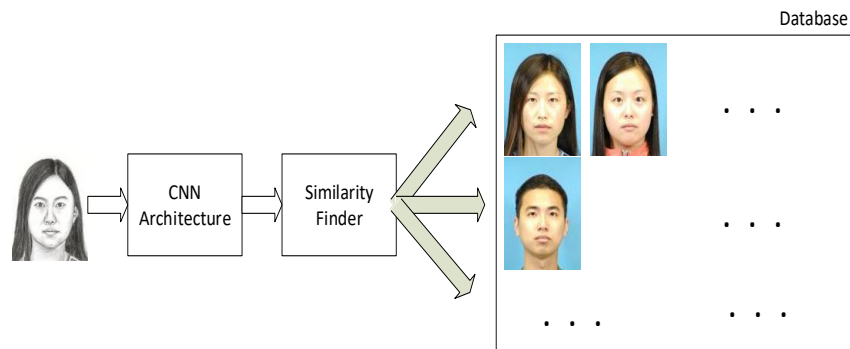


Figure 17 Sketch-Face recognition problem

The CNN architecture in Figure 18 has been proposed to solve this problem. The sketch image applied to CNN input is obtained as face photograph after 6 convolution, 6 ReLU, 4 pooling and 2 deconvolution layers.

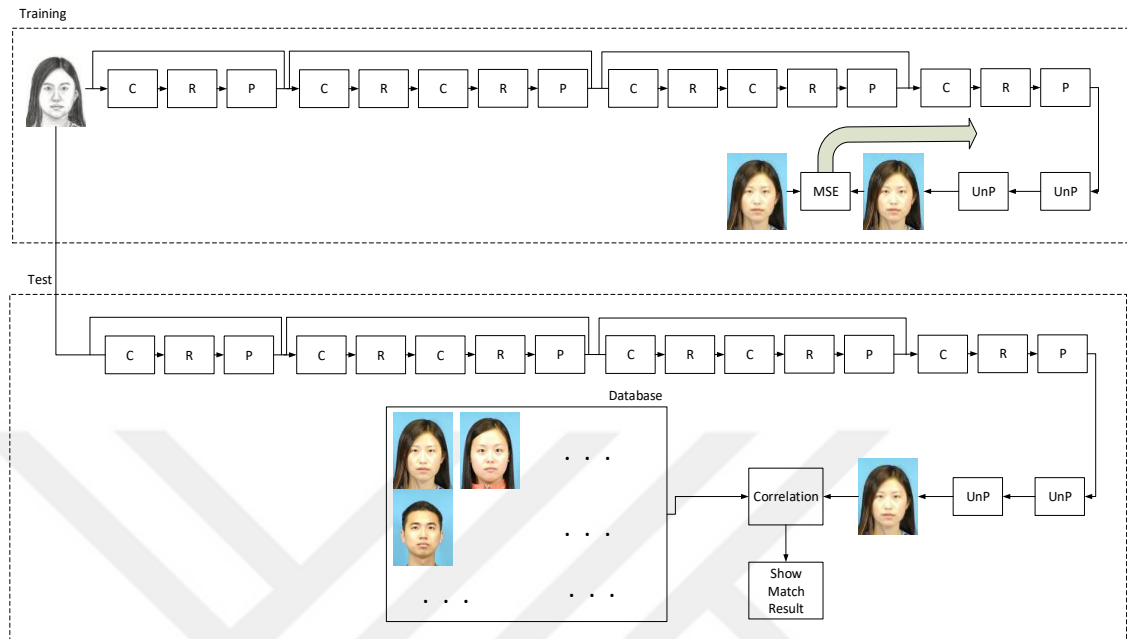


Figure 18 Proposed Method

In order for a network to be able to be trained strongly, the number of training cases must be high. The lower limit and upper limit of the number of training samples can not be predicted precisely, but the more the training sample, the lower the standard deviation. Especially the number of parameters in CNN constructions is rather high.

As the number of parameters increases, the number of training instances required must increase. Otherwise all parameters cannot reach optimum value. Nowadays, the spread of digital platforms creates a lot of data. However, these data must be labeled to be used in the training process. Tagging of images is very costly. Labeling of some images requires expert knowledge. For many medical and forensic applications, finding data is problematic. For all these reasons, the labeled data at hand should be used in the best possible way. For this, translation, deformation, reflection process etc. are applied to the training images.

In this study, the total number of images in the whole database is 188. This data needs to be divided into 3 different classes as training, validation and testing. After this

discrimination, 148 images remain for training. With so few images it is impossible to generalize the sketch-face recognition problem. Also, AR dataset used for testing process with 123 images.

Also, it is aimed to provide data diversity in the training process by producing virtual data with the data augmentation method. With the developing technology, it will not be possible to encounter augmented images that we have increased data in real life. Due to the low number of images in the data set used alone, our network may not show high performance. We need to increase our diversity in order to get rid of this problem. For this reason, training data is multiplied by 3 with two augmentation techniques. As the first augmentation technique, a gaussian noise was added to the images. For this operation, the 2D gaussian operation is applied to the image as in Equation 3.35. σ parameter in Equation 3.35 is selected as 2.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (0.35)$$

As the second data augmentation technique, the salt & pepper noise was added to all the images. For this purpose, noise is added randomly to the noise with a certain value. This noise is added to some images in color and some images in black and white. As the salt & pepper noise value, twenty-one of the total number of pixels of the view is applied. This is to create distortion in some sections without disappearing the outline of the aimed image. Thus, in the training process, generalization is achieved.



Figure 19 Image Dataset. a) original images, b) gaussian noise, c) salt-papper noise

Convolutional neural network architecture is very inspiring for the solution of image processing problems. Deep nets can often learn low, middle and high level features related to the problem. They can then automatically classify these features by themselves. Recent research has shown that the depth of the CNN network and its architect are very influential to learning [63]. But deep networks have some problems. As the network depth increases, the learning process becomes saturated. This saturation confirms the network memorization problem. In this case, as the training value continues to decrease, the validation accuracy value decreases. In some cases, the training accuracy does not decrease in the curve. More training can be used to overcome this problem. However, it is difficult to find more labeled training data. Residual structures are used to solve this problem in deep networks. In residual structures, the data at the entry of the layer is added to the last layer after processing. At this point, basic information about the data is not lost.

The proposed CNN architecture is basically a residual CNN structure. At the end of the network, the deconvolution structure is added. In this way, the original face images can be created. In the first part of the CNN structure, 6 convolution layers, 6 ReLU layers and 4 pooling layers are used. All convolution layers in the structure have 3x3 windows and the stride parameter is set to 1. Convolution is applied as in Equation 3.36 [64].

$$I_i^l = f \left(\sum_j I_j^{l-1} \otimes w_{ij}^l + b_i^l \right) \quad (0.36)$$

The task of the ReLU layer is to prevent the network from being linear. For this reason, a very simple process is applied. Network parameters with negative values are assigned as 0. The pooling layer is usually used to reduce the spatial dimensions of the image by down sampling. The max-pooling layer is used in this study. The Max-pooling layer selects the parameter with the greatest value from the pixels covered by the pooling window. At this point, the image dimensions are reduced to the window dimensions. Max-pooling is seen in Equation 3.37 [64].

$$P_{jm} = \max_{k=1}^r \left(I_{j(m-1)n+k} \right) \quad (0.37)$$

When designing new CNN architectures, the image dimensions at the output of each layer should be appropriate for the next layer. For this reason, the dimensions of the image to be formed at the exit of each layer must be calculated. Stride, padding and window size affect the size of the image to be formed. Equation 4 is used to calculate the size of the image at the output of the convolution layer. Equation 3.38 is used to calculate the viewing dimensions at the exit of the pooling layer.

$$P_c = \frac{S(Dim-1)+w-I}{2} \quad (0.38)$$

$$P_p = \frac{(w_p-1)}{2} \quad (0.39)$$

Finally, the resulting feature matrices must be reinstated. For this reason, unpooling is applied to all matrices in the network output. The task of the unpooling layer is to do the opposite of pooling. In this process, image sizes are increased according to stride, padding and window size. The generated face image is compared with the ground truth image. The network is trained with the mean square error obtained for each pixel. For this purpose stochastic gradient descent method is used.

In the test process, test images are applied to the network input in a sequence. The network produces a face photo for each sketch. The obtained image is subjected to standard single Structural Similarity Index (SSIM) process with all face photographs in the data set. By selecting the image with the highest SSIM coefficient, the detection of the possible suspect is performed. If the SSIM coefficient is below the threshold value (similarity confidence coefficient), a warning is generated that there is no such face image in the dataset.

CHAPTER 5

RESULTS

Sketch images for network training are divided into mini-batches. Training the network with a mini-batch reduces the fluctuation in the learning curve and increase the ability of generalization. For this purpose, sketch images in the dataset are applied to the network in mini-batches consisting of 10 images. The network parameters are updated with the total error for each mini-batch. All convolution layers in the proposed architecture consist of 3x3 windows. In addition, the stride values are set to 1 and the padding parameter is set to 0. In the first convolution layer, the depth of the sketch image (R, G, B) consisting of 3 layers is increased to 512. Then, this depth is increased to 1024 and reduced to 3 in the last layer. All pooling layers in the proposed CNN structure have 2x2 windows. The stride value of all pooling windows is set to 1 and padding is not applied. Increasing the stride values of the pooling layers causes further reduction of the image dimensions. This process, which is very successful for classification operations, can drastically damage the image for retrieval systems. In the unpooling section, the image is converted to its original dimensions. For this reason, the image is enlarged with two unpooling layers. Residual part is the same as classical residual structure. The information at the output of the previous layer is added to the output of the next layer.

The proposed network training and validation curves are shown in Figure 20. The training was stopped after 1000 epochs to verify training success. When the curves are examined, a steady decline indicated that the learning process was successful. Increasing the number of epochs is not likely to disturb the learning behavior of the network. From here it is possible to ascertain that the number of samples is sufficient. The steady continuation of the slope between the training and validation curves indicates that there is no memorization. A dropout of 0.3 was used in the training. Thus, each parameter of the network can be determined effectively. This is why the training curriculum is a little lower in success.

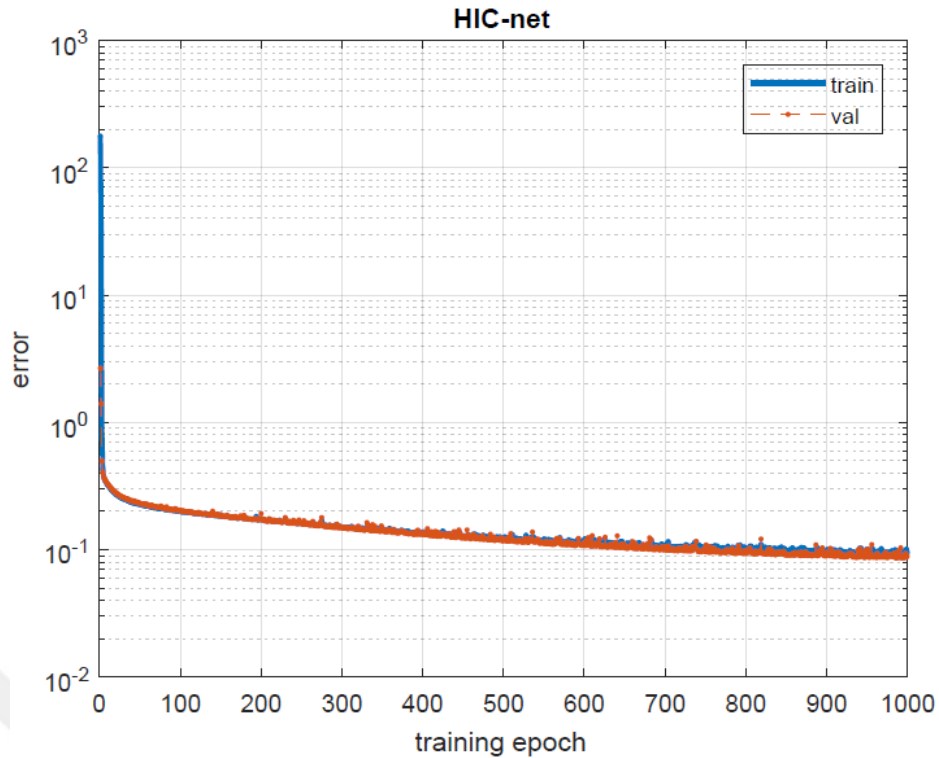


Figure 20 Training and validation curves of proposed CNN structure

According to Figure 20, the success of the training curve is 90.55% and the validation success is 91.1%. From the sketch images, the face images obtained by the proposed method are shown in Figure 21. Figure 21a shows the original sketch images. When these images are given as input to the network, the proposed network generates the face images shown in Figure 21b. When these images are examined, human faces can be perceived with outline. For color images we have just used the actual images in the data set. These are the virtual images that we have produced with the actual data increase. Due to the developments in technology, we are unlikely to meet augmented images in real life. For this reason, only color images obtained from actual data set images are visualized in the study.

In particular, basic facial features such as mouth, nose and eyebrows are quite evident. However, the eye structure is not very distinctive. To overcome this problem, an increase in the number of training samples and a deeper network structure can be offered as a solution.

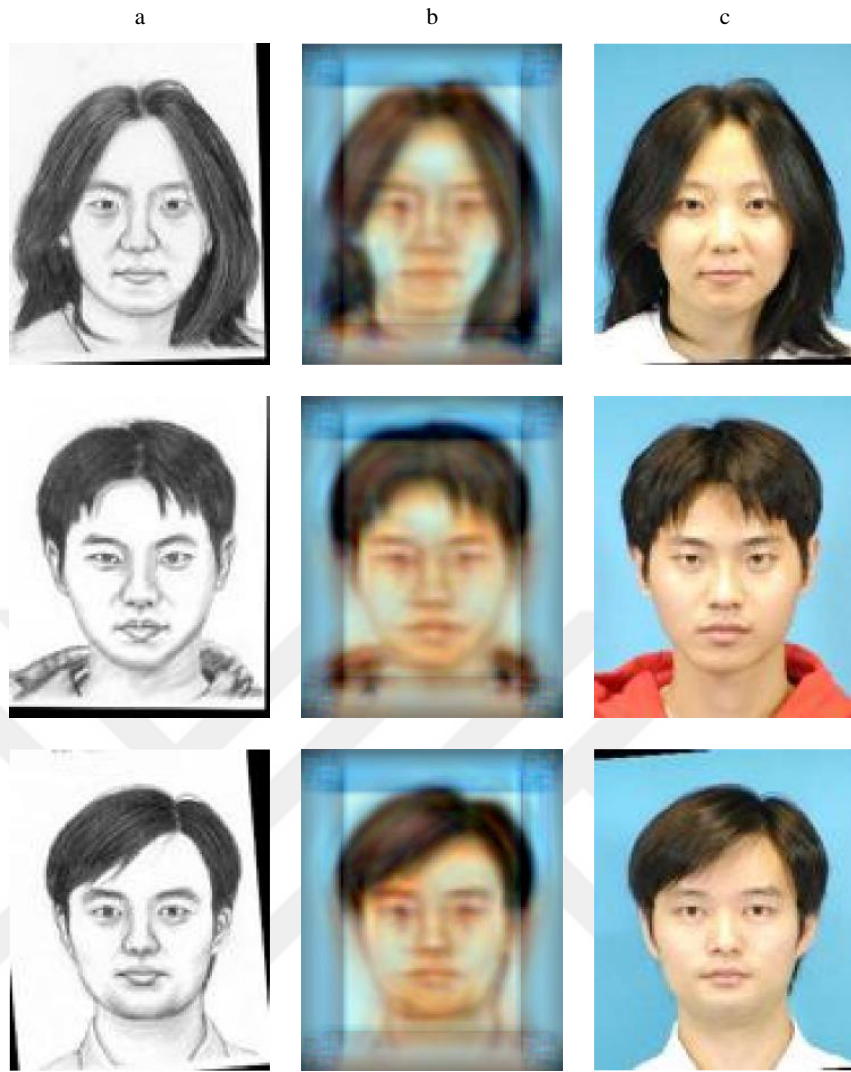


Figure 21 Face creation results of the proposed CNN architecture, a) sketch images, b) face image results of proposed CNN structures, c) ground truth face photos in CUHF dataset



Figure 22 Face creation results of the proposed CNN architecture, a) sketch images, b) face image results of proposed CNN structures, c) ground truth face photos in AR dataset

Finally, the face images generated by the proposed network should be specified in the data set. For this purpose, the relation between all the images in the produced dataset is examined. The SSIM coefficient can provide similarity information. For this purpose, SSIM processing is applied to each image in the data set. The image pair that produces the highest SSIM coefficient is matched. But here a fundamental problem arises. The image that produces the highest SSIM coefficient in the entire data set is matched with the sketch which is not a good match. A threshold value has been set in order to avoid such fundamental problems. In order to determine the SSIM threshold, the correlation between all face images produced by the CNN structure and the ground

truth face images is examined. The smallest of these SSIM values is chosen as the SSIM threshold value. If the face image produced by a sketch image is correlated with all the images in the data set and the generated SSIM value is lower than the threshold value, then we can come to the following conclusion: that person is not registered in our dataset.

The SSIM uses the distance function as the basis for similarity between statistical features between images. Unlike Corr2, the SIMM similarity function does not need to be gray level, while Corr2 uses histogram similarity as the similarity criterion. They obtain a theoretical criterion by obtaining the statistical and theoretical properties of the histogram. The Corr2 similarity criterion applies only to gray or binary images. It also cannot be used in RGB images. In this study, RGB images were obtained with RCNN structure from face sketch images. The use of SIMM and Corr2 in Binary images further reduces accuracy. Because, feature loss occurs during the conversion of RGB images to binary images.

The SSIM curve used in the determination phase is shown in Figure 23. According to Figure 23, the rate is 93.89%. SSIM is applied to the images for the face detection from the database, which is the second step of the proposed system.

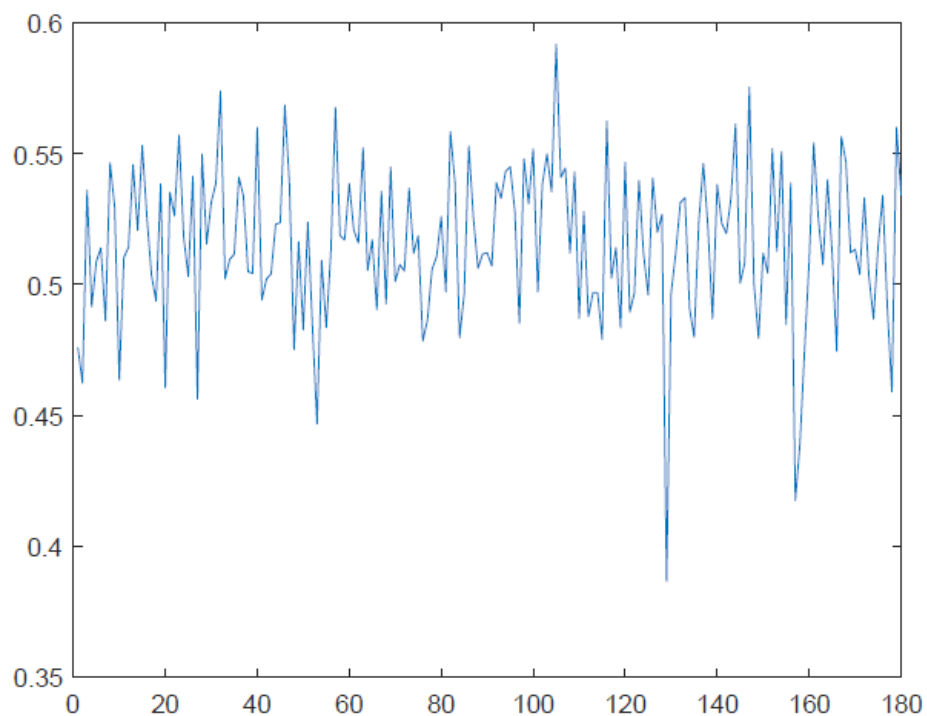


Figure 23 Face SSIM curve

Correctly determined face images and incorrectly specified face images were used to measure the success of the determination of the obtained face images in the data set. Accordingly, 176 images out of 188 images are correctly displayed. The remaining 12 face images were paired with the wrong persons.

As a result of the experiments performed, the correlation threshold value was set at 70. The correlation curve used in the determination phase is shown in Figure 24. According to Figure 24, when the threshold value is set to 70, the error rate is 5.5%. Correlation is applied to the images for the face detection from the database, which is the second step of the proposed system.

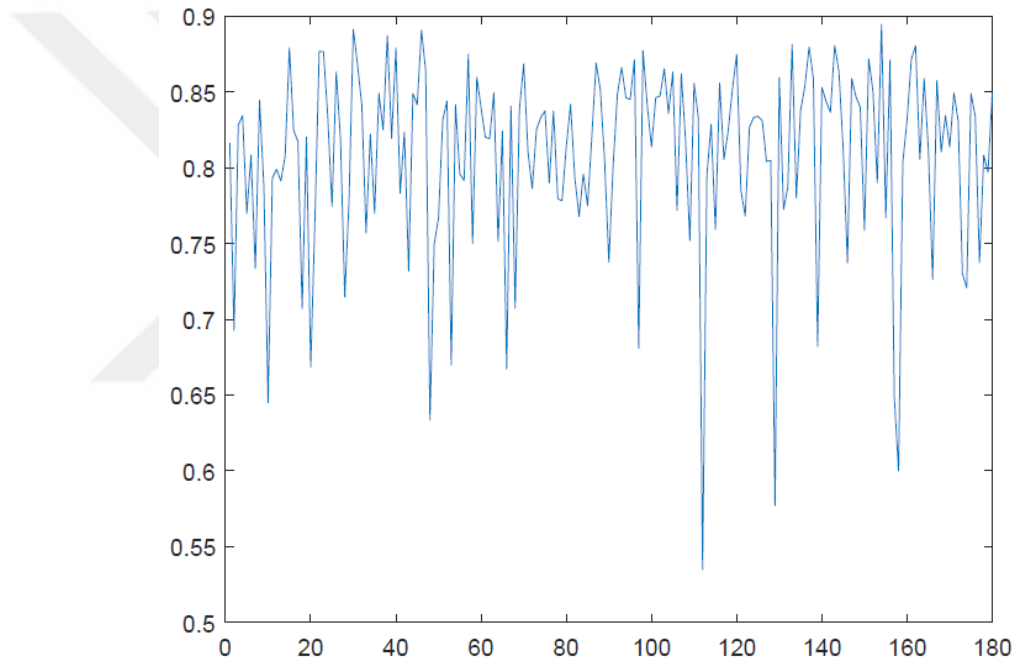


Figure 24 Face correlation curve

Correctly determined face images and incorrectly specified face images were used to measure the success of the determination of the obtained face images in the data set. Accordingly, 143 images out of 188 images are correctly displayed. The remaining 45 face images were paired with the wrong persons. In this case, the matching performance of the proposed method is roughly determined as 76.06%.

CHAPTER 6

CONCLUSION

In this thesis, face images from a sketch image were detected in a dataset. For this purpose, a CNN structure with residual architecture is proposed. The deconvolution unit is added to the output layer of the proposed CNN structure so that the original image dimensions can be obtained. Finally, the correct person in the dataset is determined by the SSIM method. In the first step of the proposed method, color face images are obtained from sketch images. The training and validation success rates in this section are quite high. The proposed network structure has passed a successful training process even though the number of labeled samples is limited. In the second part, the SSIM coefficient is used in the image matching part of the dataset. In future studies, the proposed network structure will be improved.

APPENDICES

MAIN CODE

```
setup();  
  
setup('useGpu', true);  
  
names_gt = dir('sira/*.png');  
  
names_gt = fullfile('sira', {names_gt.name});  
  
names_gt = dir('karsi/*.png');  
names_gt = fullfile('karsi', {names_gt.name});  
  
batch_sayisi=1;  
batch_boyutu=180;  
batch_boyutu_tam=180;  
batch_dondurme=1;  
iterasyon_sayisi=200;  
  
for ii=1:batch_sayisi  
  
k=180;  
r1 = randperm(k,(batch_boyutu));  
names1=names_gt(r1);  
names1_gt=names_gt(r1);
```

```

sayac1=0;

for i=1:1:batch_boyutu_tam

    sayac1=sayac1+1;

    im_batch(:,:,i)=imread(names1{sayac1});

    labels(:,:,i)=imread(names1_gt{sayac1});

    imdb.images.set(i)=1;

    imdb.images.id(i)=i;

end

im_batch=mat2gray(im_batch);

im_batch=single(im_batch);

im_labels=mat2gray(labels);

im_labels=single(im_labels);

end

imdb.images.set((batch_boyutu_tam-
(batch_boyutu_tam/5)+1):(batch_boyutu_tam))=2;

imdb.images.label=im_labels;

imdb.images.data=im_batch;

net = initializeLargeCNN() ;

net = addCustomLossLayer(net, @l2LossForward, @l2LossBackward) ;

trainOpts.expDir = 'data/text-small' ;

trainOpts.gpus = [1] ;

trainOpts.batchSize = 1 ;

trainOpts.learningRate = 0.0000001 ;

```

```
trainOpts.plotDiagnostics = false ;  
trainOpts.numEpochs = 1000 ;  
trainOpts.errorFunction = 'multiclass' ;  
net = cnn_train(net, imdb, @getBatch, trainOpts) ;
```



STRUCTURE CODE

```
function net = initializeLargeCNN()
```

```
net.layers = { } ;
```

```
net.layers{end+1} = struct(...
```

```
    'name', 'conv1', ...
```

```
    'type', 'conv', ...
```

```
    'weights', {xavier(3,3,3,256)}, ...
```

```
    'pad', 0, ...
```

```
    'learningRate', [1 1], ...
```

```
    'weightDecay', [1 0]) ;
```

```
net.layers{end+1} = struct(...
```

```
    'name', 'relu1', ...
```

```
    'type', 'relu') ;
```

```
net.layers{end+1} = struct('type', 'pool', ...
```

```
    'method', 'max', ...
```

```
    'pool', [3 3], ...
```

```
    'stride', 1, ...
```

```
    'pad', 0) ;
```

```
net.layers{end+1} = struct('type', 'concat', ...
```

```
'method', 'net.layer{1}','net.layer{3}';
```

```
net.layers{end+1} = struct(...  
    'name', 'conv2', ...  
    'type', 'conv', ...  
    'weights', {xavier(3,3,256,256)}, ...  
    'pad', 0, ...  
    'learningRate', [1 1], ...  
    'weightDecay', [1 0]);
```

```
net.layers{end+1} = struct(...  
    'name', 'relu2', ...  
    'type', 'relu');
```

```
net.layers{end+1} = struct(...  
    'name', 'conv3', ...  
    'type', 'conv', ...  
    'weights', {xavier(3,3,256,256)}, ...  
    'pad', 0, ...  
    'learningRate', [1 1], ...  
    'weightDecay', [1 0]);
```

```
net.layers{end+1} = struct(...  
    'name', 'relu3', ...
```



```
'type', 'relu');
```

```
net.layers{end+1} = struct('type', 'pool', ...  
    'method', 'max', ...  
    'pool', [3 3], ...  
    'stride', 1, ...  
    'pad', 0);
```

```
net.layers{end+1} = struct('type', 'concat', ...  
    'method', 'net.layer{3}', 'net.layer{8}');
```

```
net.layers{end+1} = struct(...  
    'name', 'conv2', ...  
    'type', 'conv', ...  
    'weights', {xavier(3,3,256,256)}, ...  
    'pad', 0, ...  
    'learningRate', [1 1], ...  
    'weightDecay', [1 0]);
```

```
net.layers{end+1} = struct(...  
    'name', 'relu2', ...
```

```
'type', 'relu') ;
```

```
net.layers{end+1} = struct(...  
    'name', 'conv3', ...  
    'type', 'conv', ...  
    'weights', {xavier(3,3,256,256)}, ...  
    'pad', 0, ...  
    'learningRate', [1 1], ...  
    'weightDecay', [1 0]) ;
```

```
net.layers{end+1} = struct(...  
    'name', 'relu3', ...  
    'type', 'relu') ;
```

```
net.layers{end+1} = struct('type', 'pool', ...  
    'method', 'max', ...  
    'pool', [3 3], ...  
    'stride', 1, ...  
    'pad', 0) ;
```

```
net.layers{end+1} = struct('type', 'concat', ...  
    'method', 'net.layer{8}','net.layer{13}') ;
```

```
net.layers{end+1} = struct(...
```

```
'name', 'conv3', ...
'type', 'conv', ...
'weights', {xavier(3,3,256,256)}, ...
'pad', 0, ...
'learningRate', [1 1], ...
'weightDecay', [1 0]);
```

```
net.layers{end+1} = struct(...
    'name', 'relu3', ...
    'type', 'relu');
```

```
net.layers{end+1} = struct('type', 'pool', ...
    'method', 'max', ...
    'pool', [3 3], ...
    'stride', 1, ...
    'pad', 0);
```

```
net.layers{end+1} = struct('type', 'unpool', ...
    'method', 'max', ...
    'pool', [(size(net.layer(1,1)/2)) (size(net.layer(1,1)/2))], ...
    'stride', 1, ...
    'pad', 0);
```

```
net.layers{end+1} = struct('type', 'unpool', ...  
    'method', 'max', ...  
    'pool', [(size(net.layer(1,1)/2)) (size(net.layer(1,1)/2)], ...  
    'stride', 1, ...  
    'pad', 0);
```

```
net = vl_simplenn_tidy(net);
```

SSIM CODE

```
names_beyaz1 = dir('sonuc/*.png') ;
names_beyaz1 = fullfile('sonuc', {names_beyaz1.name}) ;

names_beyaz_gt1 = dir('karsi/*.png') ;
names_beyaz_gt1 = fullfile('karsi', {names_beyaz_gt1.name}) ;

for kkl=1:180

    for kkk=1:180
        sonucum=imread(names_beyaz1{kkl});
        karsm=imread(names_beyaz_gt1{kkk});
        sonucum=rgb2gray(sonucum);
        karsm=rgb2gray(karsm);
        noktalar(kkk)=ssim(sonucum,karsm);
    end

    [basari(kkl) indisi(kkl)]=max(noktalar);
end

yuzde=0;
for kkk=1:180

    if (indisi(kkk)==kkk)
```

```
yuzde=yuzde+1;  
end
```

```
end
```

```
yuzde=yuzde/180;
```

CORR2 CODE

```
names_beyaz1 = dir('sonuc/*.png') ;  
names_beyaz1 = fullfile('sonuc', {names_beyaz1.name}) ;  
  
names_beyaz_gt1 = dir('karsi/*.png') ;  
names_beyaz_gt1 = fullfile('karsi', {names_beyaz_gt1.name}) ;  
  
for kkl=1:180  
  
    for kkk=1:180  
  
        sonucum=imread(names_beyaz1{kkl});  
        karsm=imread(names_beyaz_gt1{kkk});  
        sonucum=rgb2gray(sonucum);  
        karsm=rgb2gray(karsm);  
        noktalar(kkk)=corr2(sonucum,karsm);
```

```
end

[basari(klk1) indisi(klk1)]=max(noktalar);

end

yuzde=0;

for kkk=1:180

    if (indisi(kkk)==kkk)

        yuzde=yuzde+1;

    end

end

yuzde=yuzde/180;
```

REFERENCES

- [1] N. Lavanyadevi, S. Priya and K. Krishanthana, "Performance Analysis of Face Matching and Retrieval In Forensic Applications," *International Journal of Advanced Electrical and Electronics Engineering*, vol. 2, pp. 100-104, 2013.
- [2] A. K. Jain, B. Klare and U. Park, "Face Matching and Retrieval in Forensics Applications," *IEEE Computer Society*, vol. 12, pp. 21-27, 2012.
- [3] A. K. Jain, B. Klare and U. Park, "Matching forensic Sketches and Mug Shots to Apprehend Criminals," *IEEE Computer Society*, vol. 12, pp. 95-96, 2011.
- [4] A. R. Sharma and P. R. Devale, "An Application to Human Face Photo- Sketch Synthesis and Recognition," *International Journal of Advances in Engineering & Technology*, vol. 2, pp. 55-64, 2012.
- [5] X. Wang and X. Tang, "Face photo-sketch synthesis and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 31, no. 11, pp. 1955-1967, 2009.
- [6] K. Bonnen, B. F. Klark and A. K. Jain, "Component Based Representation in Automated Face Recognition," *IEEE Transactions on Information Forensics And Security*, vol. 8, pp. 239-253, 2013.
- [7] FISWG, "Facial Identification Scientific Working Group," 15 June 2009. [Online]. Available: <https://fiswg.org/index.htm>. [Accessed 23 May 2017].
- [8] ENFSI, "European Network of Forensic Science Institutes," 2017. [Online]. Available: <http://www.enfsi.eu/>. [Accessed 23 May 2017].
- [9] NICFS, "National Institute of criminology and forensic science," 2017. [Online]. Available: <http://enfsi.eu/>. [Accessed 23 May 2017].
- [10] D. Mcquiston, L. Topp and R. Malpass, "Use of facial composite systems in US law enforcement agencies," *Psychology, Crime and Law*, vol. 12, no. 5, pp. 505-517, 2006.
- [11] X. Wang and X. Tang, "Face sketch recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 40, no. 1, pp. 50-57, 2004.
- [12] Q. Liu, X. Tang, H. Jin, H. Lu and S. Ma, "A nonlinear approach for face sketch synthesis and recognition.," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, 2005.
- [13] B. Klare, Z. Li and A. Jain, "Matching forensic sketches to mug shot photos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 639-646, 2011.
- [14] D. Lowe, "Distinctive image features from scale-invariant key points,," *Int. J. Computer. Vis.*, vol. 60, no. 2, pp. 91-110, 2004.

- [15] T. Ojala, M. Pietikainen and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 24, no. 7, pp. 971-987, 2002.
- [16] B. Heisele, P. Ho, J. Wu and T. Poggio, "Face recognition: Component- based versus global approaches," *Comput. Vis Image Understand*, vol. 91, no. 1, pp. 6-21, 2003.
- [17] Y. Ivanov, B. Heisele and T. Serre, "Using component features for face recognition," in *Proc. Automatic Face and Gesture Recognition*, Ljubljana, 2004.
- [18] Lin and X. Tang, "Recognize high resolution faces: From macrocosm to microcosm," 2006.
- [19] B. Klare, A. Paulino and A. Jain, "Analysis of facial features in identical twins," in *Proc. Int. Joint Conf. Biometric*, Colorado, 2011.
- [20] J. Huang, V. Blanz and B. Heisele, "Face recognition using component- based SVM classification and morphable models," in *Proc. First Int. Workshop on Pattern Recognition with Support Vector Machines*, Niagara Falls, 2002.
- [21] A. J. Goldstein, L. D. Harmon and A. B. Lesk, "Identification of Human Faces," *Proc. IEEE*, vol. 59, no. 5, pp. 748-760, 1971.
- [22] L. Sirovich and M. Kirby, "A Low-Dimensional Procedure for the Characterization of Human Faces," *J. Optical Soc. Am. A*, , vol. 4, no. 3, pp. 519-524, 1987.
- [23] M. A. Turk and A. P. Pentland, "Face Recognition Using Eigenfaces," *Proc. IEEE*, pp. 586-591, 1991.
- [24] W. Zhang, X. Wang and X. Tang, "Coupled Information-Theoretic Encoding for Face Photo-Sketch Recognition," in *Proc. Conf. Computer Vision and Pattern Recognition*,, 2011.
- [25] X. Gao, J. Zhong, J. Li and C. Tian, "Face Sketch Synthesis Algorithm Based on E-HMM and Selective Ensemble," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 18, no. 4, pp. 487-496, 2008.
- [26] H. Nizami, J. Adkins-Hill, Y. Zhang, J. Sullins, C. McCullough, S. Canavan and L. Yin, "A biometric database with rotating head videos and hand-drawn face sketches," in *Proc. of IEEE Conference on Biometrics: Theory, Applications and Systems*, 2009.
- [27] B. Klare and A. Jain, "Sketch to photo matching: A feature-based approach," in *Proc. SPIE Conference on Biometric Technology for Human Identification*, 2010.
- [28] A. Sharma and D. Jacobs, "Bypassing Synthesis: PLS for Face Recognition with Pose, Low-Resolution and Sketch," i," in *Proc. IEEE Conf Computer Vision and Pattern Recognition*, 2011.

- [29] Z. Lei and S. Li, "Coupled Spectral Regression for Matching Heterogeneous Faces," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009.
- [30] M. Ahmed and F. Bobere, "Criminal Photograph Retrieval based on Forensic Face Sketch using Scale Invariant Feature Transform," in *International conference on Technology and Business Management*, 2012.
- [31] H. S. Bhatt, S. Bharadwaj, R. Singh and M. Vatsa, "Memetically Optimized MCWLD for Matching Sketches with Digital Face Images," *IEEE Transactions on Information Forensics And Security*, vol. 7, pp. 1522-1535, 2012.
- [32] U. Park and A. K. Jain, "Face Matching and Retrieval using Soft Biometrics," *IEEE Transactions on Information Forensics And Security*, vol. 3, pp. 406-415, 2010.
- [33] P. Viola and M. J. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern IEEE Computer Society Conference on Computer Vision and Pattern*, 2001.
- [34] Z. Tabatabaie, R. Rahmat, N. Udzir and E. Kheirkhah, "A hybrid face detection system using combination of appearance-based and feature-based methods," *IJCSNS International Journal of Computer Science and Network Security*, vol. 9, no. 5, 2009.
- [35] S. Wang and A. Abdel-Dayem, "Improved Viola-Jones Face Detector," Laurentian University Sudbury, Ontario, 2014.
- [36] Y.-W. Wu, "Face Detection in Color Images Using AdaBoost Algorithm Based on Skin Color Information," Huazhong Normal University, Wuhan, 2008.
- [37] M. Niazi and S. Jafari, "Hybrid face detection in color images," *IJCSI International Journal of Computer Sciences Issues*, vol. 7, pp. 367-373, 2010.
- [38] C. Erdem, S. Ulukaya, A. Karaali and A. Erdem, "Combining haar feature and skin color based classifiers for face detection," in *IEEE 36th International Conference on Acoustics, Speech and Signal Processing*, 2011.
- [39] N. J. Butko, L. Zhang, G. W. Cottrell and J. R. Movellan, "Visual Saliency Model for Robot Cameras," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2008.
- [40] L. Itti, C. Koch and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254-1259, 1998.
- [41] A. Shahab, F. Shafait and A. Dengel, "How Salient is Scene Text?," in *IEEE Conference on Document Analysis Systems (DAS)*, 2012.

- [42] J. Harel, C. Koch and P. Perona, “Graph-based visual saliency,” in *Advances in Neural Information Processing Systems*, Cambridge, MIT Press, 2007.
- [43] L. Zhang, M. H. Tong, T. K. Marks, H. Shan and Cottrel, “A Bayesian framework for saliency using natural statistics,” *Journal of Vision*, vol. 8, no. 7, pp. 1-20, 2007.
- [44] Jin, K. H., McCann, M. T., Froustey, E., & Unser, M. (2017). Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9), 4509-4522.
- [45] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).
- [46] Tang, X., & Wang, X. (2004). Face sketch recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1), 50-57.
- [47] Liu, Q., Tang, X., Jin, H., Lu, H., & Ma, S. (2005, June). A nonlinear approach for face sketch synthesis and recognition. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (Vol. 1, pp. 1005-1010). IEEE.
- [48] Zhang, W., Wang, X., & Tang, X. (2011, June). Coupled information-theoretic encoding for face photo-sketch recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (pp. 513-520). IEEE.
- [49] Galoogahi, H. K., & Sim, T. (2012, July). Inter-modality face sketch recognition. In *Multimedia and Expo (ICME), 2012 IEEE International Conference on* (pp. 224-229). IEEE.
- [50] Mittal, P., Vatsa, M., & Singh, R. (2015, May). Composite sketch recognition via deep network-a transfer learning approach. In *Biometrics (ICB), 2015 International Conference on* (pp. 251-256). IEEE.
- [51] Zhu, Z., Luo, P., Wang, X., & Tang, X. (2014). Recover canonical-view faces in the wild with deep neural networks. *arXiv preprint arXiv:1404.3543*.
- [52] X. Tang and X. Wang, Face Sketch Recognition, *IEEE Transactions On Circuits And Systems For Video Technology*, Vol. 14, No. 1, January 2004.
- [53] K. Bansode N, and Sinha P K. “Face Sketch Generation Using Evolutionary Computing.” *International Journal on Soft Computing*, vol. 7, no. 4, 2016, pp. 01–10., doi:10.5121/ijsc.2016.7401.
- [54] S. Pramanik and D. Bhattacharjee, *An Approach: Modality Reduction and Face-Sketch Recognition*, 2013, ArXiv.
- [55] Simon Haykin, *Neural Networks and Learning Machines*, (Pearson Education, United States of America, 2009).
- [56] David Kriesel, *A Brief Introduction to Neural Networks*, (2005).

- [57] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning(Adaptive Computation and Machine Learning series), (MIT Press, Cambridge, 2016).
- [58] Christos Stergiou, Dimitrios Sigano, Neural Networks, Imperial College London CS11 Notes, (1996).
- [59] James L. McClelland, Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises, (Standford University, 2015).
- [60] Jeffrey L. Elman, Finding Structure in Time, Cognitive Science 14, (1990).
- [61] Li Deng, Dong Yu Now, Deep Learning: Methods and Applications, (Now, Publisher, Boston USA, 2014).
- [58] Larry Brown, Deep Learning With Gpus, Geoint Sunum, (2015).
- [59] James L. McClelland, Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises, (Standford Üniversitesi, 2015).
- [60] Jeffrey L. Elman, Finding Structure in Time, Cognitive Science 14, (1990).
- [61] Li Deng, Dong Yu Now, Deep Learning: Methods and Applications, (Now, Publisher, Boston USA, 2014).
- [62] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research 15, (2014).
- [63] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
- [64] Öztürk, Ş., & Akdemir, B. A convolutional neural network model for semantic segmentation of mitotic events in microscopy images. Neural Computing and Applications, doi: 10.1007/s00521-017-3333-9.
- [65] Zhang, L., Lin, L., Wu, X., Ding, S., & Zhang, L. (2015). End-to-End Photo-Sketch Generation via Fully Convolutional Representation Learning. Proceedings of the 5th ACM on International Conference on Multimedia Retrieval - ICMR 15. doi:10.1145/2671188.2749321.