**DOKUZ EYLÜL UNIVERSITY**
**GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

# MOOD ANALYSIS OF EMPLOYEES BY USING IMAGE-BASED DATA

**by**
**Özgür Deniz GÜNSELİ**

**October, 2019**
**İZMİR**

# MOOD ANALYSIS OF EMPLOYEES BY USING IMAGE BASED DATA

**A Thesis Submitted to the**
**Graduate School of Natural and Applied Sciences of Dokuz Eylül University**
**In Partial Fulfillment of the Requirements for the Degree of Master of Science**
**in Computer Engineering, Computer Engineering Program**

**by**
**Özgür Deniz GÜNSELİ**

**October, 2019**
**İZMİR**

**M.Sc THESIS EXAMINATION RESULT FORM**

We have read the thesis entitled **"MOOD ANALYSIS OF EMPLOYEES BY USING IMAGE-BASED DATA"** completed by **ÖZGÜR DENİZ GÜNSELİ** under supervision of **ASST. PROF. DR. SEMİH UTKU** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

.....
Asst. Prof. Dr. Semih UTKU

Supervisor

.....
Doç.Dr. Mehmet Hilal Özcanhan

Jury Member

.....
Prof. Dr. Mehmet İnlütek

Jury Member

Prof.Dr. Kadriye ERTEKİN

Director

Graduate School of Natural and Applied Sciences

**ACKNOWLEDGMENTS**

First of all, I would like to thank my advisor Assistant Professor Dr. Semih UTKU for his help, suggestions and guidance to this study.

Also, I owe my colleagues from Metadata Bilişim Teknoloji Sanayi ve Tic. A.Ş. a thank you for the support they have shown in this thesis working process.

I sincerely thank my wife Firdevs Gümüş Günseli for her understanding, patience and support in the whole process and for her support in all matters.

I would like to thank my brother Imran Azat GÜNSELİ and my cousin İsmail Gökhan GÜNSELİ, who we supported each other till the late hours while dealing with our theses.

Özgür Deniz GÜNSELİ

**MOOD ANALYSIS OF EMPLOYEES BY USING IMAGE-BASED DATA**

## ABSTRACT

Today, artificial intelligence and machine learning are used in many fields. Artificial intelligence and machine learning methods are used for many jobs that can be conducted with human intelligence and perception. In recent years, these approaches have been used frequently especially in object perception and classification. With the development of the techniques and equipment used, the success rate increases rapidly.

Face detection and face detection is an area where these technologies are frequently used. Almost all the applications and web sites we use this technology are encountered. In addition to long-term technologies such as face detection from picture or from video, technologies such as face recognition are also used for security purposes. Facial recognition and face detection are included in our daily lives with many similar areas.

The thesis is aimed to design a solution that detects the emotions of people through visual data by using face detection and face recognition technologies using a method free of personal data and reporting them individually or in groups.

In order to provide a solution, a general system was developed to analyze the emotion on the face data that was drawn into the scope of the thesis. With this system, emotion analysis is conducted and reporting is provided by using artificial intelligence methods based on personal data collected from terminals.

**Keywords**: Face detection, face recognition, artificial intelligence, convolutional neural networks, image processing

# GÖRÜNTÜ TABANLI VERİ KULLANARAK ÇALIŞANLARIN DUYGU ANALİZİ

## ÖZ

Günümüzde yapay zeka ve makine öğrenmesi birçok alanda kullanılmaktadır. İnsan zekası ve algısıyla yürütülebilecek birçok iş için yapay zeka ve makine öğrenmesi yöntemleri kullanılmaktadır. Son yıllarda özellikle nesne algılaması ve sınıflandırması konusunda sıklıkla bu yaklaşımlardan yararlanılmaktadır. Kullanılan teknik ve donanımların gelişmesi ile birlikte başarı oranı da hızlıca artmaktadır.

Yüz algılama ve yüz tanıma, bu teknolojilerin sıklıkla kullanıldığı bir alandır. Hemen hemen kullandığımız tüm uygulamalarda ve web sitelerinde bu teknolojiyle karşılaşmaktayız. Resim üzerinden yüz algılama, videolardan yüz algılama gibi uzun zamandır kullanılan teknolojilerin yanı sıra, güvenlik amacıyla yüz tanıma gibi teknolojiler de kullanılmaktadır. Buna benzer birçok alanla birlikte yüz tanıma ve yüz algılama, günlük hayatımızın içerisinde yer almaktadır.

Bu tez, kişisel verilerden arındırılmış bir yöntem ile, yüz algılama ve yüz tanıma teknolojilerini kullanarak görsel veriler üzerinden insanların ruhsal durumlarını algılayarak, bunları tekil ya da grupsal olarak raporlayacak bir çözüm sunmayı hedeflemektedir.

Çözüm üretmek amacıyla, tez kapsamında çizim haline getirilmiş yüz verileri üzerinden ruhsal durum analiz etmeye yönelik genel bir sistem oluşturuldu. Oluşturulan bu sistem ile, uçbirimlerden toplanan kişisel verilerden arındırılmış veriler üzerinden yapay zeka yöntemleri kullanılarak ruhsal durum analizi yapılmakta ve raporlama sağlanmaktadır.

**Anahtar kelimeler**: Yüz algılama, yüz tanıma, yapay zeka, evrişimli sinir ağları, görüntü işleme

**CONTENTS**

**Page**

**LIST OF FIGURES**

**Page**

## LIST OF TABLES

**Page**

# CHAPTER ONE
# INTRODUCTION

With industrialization rapidly growing and changing its form in recent years, enterprises evolved from workplaces with 1-9 workers to major establishments with thousands of employees. This change experienced in manufactures and services increased the number of workplaces that includes many employees working in different locations.

While this change made it increasingly difficult for firms to keep track of their employees, advancements in technology provided different solutions in this area. A number of softwares used in human resources departments of firms make it possible to analyze work processes and efficiencies of workers.

Though these human resources softwares follow many processes about employees, they are far from being able to provide any information to be user in analysis of their mental state; one of the most important determinants of efficiency. Technical and hardware advancements that we achieved in recent years on image processing and machine learning gives many opportunities for analysis by image data.

## 1.1 Overview

When we need to distinguish human from the simplest or even more advanced living beings, we use the ability to think as the main criterion. Even though neural system exists in most creatures, the way it operates in humankind is different from others. Human brain, which we are still unable to fully discover, operates in a much more advanced fashion than other living things on our planet. But when we bring it down to most basic principles, physical structure of our brain works in a similar way with much less intelligent creatures.

Neurophysiologist Warren S. McCulloch and mathematician Walter H. Pitts, who wrote a paper on work structure of neurons in 1948, developed a neural network model

using electric circuits (Mcculloch, Lettvin, Pitts, & Dell, 1952). With advancement of computers it became possible for neural network model made with electric circuits to be run on computers.

In 1951, Marvin Minsky and Dean Edmonds developed the first neural network machine with ability to learn (Ramos, Augusto, & Shapiro, 2008). Neural network machine, which had 40 neurons and was named SNARC, could set weightings of its synapses and it was successfully trained to solve a maze game.

In 1959, Arthur Samuel, one of the pioneers of machine learning, developed a computer program that could learn to play chess (Samuel, 2000). The developed program was able to analyze the current board status, plan moves and learn specific game strategies.

Studies rapidly increased on machine learning technique, which, according to Tom Mitchell, is defined as "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." (El Naqa & Murphy, 2015). Machine learning, which saw many different approaches near 2000, started to march towards modern machine learning applications with computer technology advancing and spreading in early 2000s.

Just like it is used practically in many different areas, machine learning can be used for detection of different objects on image and video data. Object detection, which can be carried out in different ways, helps us to solve real world problems in ever increasing number of areas. Today, we can, by using varied algorithms, recognize human faces on images and videos and furthermore, by analyzing the faces we recognize we can detect emotions of humans in images.

## 1.2 The Goal of the Thesis

Machine learning describes computer being trained with specific data and learning on methods to solve a specific problem. Can be used for goals like analysis of a price, detection of a dialogue providing suggestions on a particular area.

Likewise, by using machine learning, process of detecting human on picture and video data can be realised and analysis can be made on mental states of these detected people. Thanks to this process, which can be carried out in real time, mental states of people in particular region can be continuously analysed.

Goal of this thesis is to develop an application that, by using machine learning techniques, can process images coming from a camera, detect people within and analysing their mental states; so that firms can measure worker efficiency.

## 1.3 Thesis Organization

This thesis comprises of five chapters. A simple definition of face recognition on image is given in Chapter 1. In Chapter 2, works related with face recognition are examined. Chapter 3 describes methods that will be used in the thesis. Chapter 4 contains the solution which came from the study. Chapter 5 consists of final results.

# CHAPTER TWO
# RELATED WORKS

For nearly 100 years, numerous studies have been carried out on parameters that correlate with employee productivity. Many studies have demonstrated that job productivity has been positively correlated with job satisfaction (Zelenski, Murphy, & Jenkins, 2008). In addition, many studies have reported that negative and positive mental states have an effect on employee productivity (Fisher & Noble, 2004). These studies have shown that workers with negative mental status have lower job productivity. On the contrary, positive mental states increase work efficiency. It has been shown that positive mental states have a significant correlation with job productivity as well as total life satisfaction.

Another issue related to employee mental state is the tendency to change jobs. The general satisfaction of the employee with the work affects the employee's job change (Wright & Bonett, 2007; Abraham, 1999). In parallel, positive and negative moods also affect their tendency to change jobs. Employee change creates costs for large companies, especially when trained, experienced and competent employees leave their jobs (Hinkin & Tracey, 2000). Every new employee has many direct or indirect costs, such as hiring, training and wrong recruitment. These costs can reach very high values, especially in large companies.

In 2005 Lesiuk (Lesiuk, 2005) and in 2010 Miner and Glomb (Miner & Glomb, 2010) conducted work on the correlations of the mental state of the employees with the completion of work pieces, job quality and job productivity. When these studies carried out with different sample groups are examined in detail, it is seen that there is a correlation between the employee's mental status and psychological well-being and work efficiency.

In order to perform the mental state analysis processes of employees, it is necessary to standardize the mental status definitions. At this point, many different studies have been carried out regarding the standard definitions of mental states for the perception of personal mental states. In 1970 and 1971, Ekman conducted studies and identified

six basic mental states (Ekman, 1970; Ekman & Friesen, 1971). In his study in 1992, he expanded the basic mental states of people to 11 different situations (Ekman, 1992). Plutchik carried out a study involving 8 different basic mental states and their different levels and interactions in 1980 (Plutchik, 1980).

In 1967 and 1969, Ekman and Friesen showed that there is a relationship between the muscles in the face and certain mental states (Ekman & Friesen, 1969, 1971). In 1971, they published their "Facial Action Coding System" paper (Ekman & Friesen, 1971). In this publication, it is stated that different psychological conditions occur as specific facial expressions on the lips, eyes, eyebrows and cheeks and for each facial expression that occurs on the human face, "Action Unit" is defined. It is stated that each emotion causes different Action Units on human face. Many studies have been conducted on this study in different fields. Different methods and techniques have been used to extract and define these features from visual data.

There are many different methods and studies about process of face recognition on images. A number of related studies has been made in order to create different detection processes for faces over an image file. When they are examined, the procedure in processes of face and mental state detection over any image or video is processed as three steps that comprises of combinations of face detection and face recognition (Bhele & Mankar, 2012).


Figure 2.1 Steps of face recognition process

In face recognition operations, first stage is to establish whether there is a face in a particular image. In this stage, different approaches are made use of to check existence of a face in image and if there is one or more in visual, they are located within the image. Factors like posture of face, lighting and angle of image affect success rate of the method in this stage.

In second stage of face recognition process, particular attributes for the detected face are sorted out over the image, in accordance with the necessities. With different methods, particular attributes of face are numerically picked over media file. In the last stage, the classification process is carried out on these extracted features and the class of the face is determined in the structure of the desired classification structure. In some approaches, stages 2 and 3 are carried out in the same process.

## 2.1 Face Detection

Face detection is the first stage of face recognition. In this stage, faces in the visual are separated from background visual and each face is separately located. Just as human face can be came across in many different types, it also can exist in visuals in many different forms and features. Because of this, operation that will be used should be able to carry out face detection processes free of factors like light, image size and posture. At this stage several different studies are made ranging from quite simple ones to very advanced and complex ones.

### *2.1.1 Viola and Jones Algorithm*

This object detection method, which was proposed in 2001 by Paul Viola and Michael Jones, not only enables detection of any object in image, it also is used with purpose of detecting faces that are fully included in image, taken from a frontal perspective and not reversed (Viola & Jones, 2001).

In Viola and James method, in order to detect any object within the image, Haar based attributes are utilised (Lienhart & Maydt, 2002). This attributes, which are consisted of multiple 1X1 arrays, contains cells, of which some represents bright areas, and some, dark. While dark cells are assigned positive values, bright cells are assigned negative ones and the value of area is found by summation of these. Each specified Haar attribute is searched on image with sliding window method. Structures that form

contrast; like eye, nose and forehead can be detected in any image thanks to these attributes.

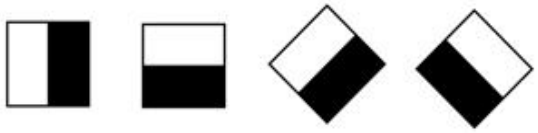


Figure 2.2 Edge features of Viola & Jones method

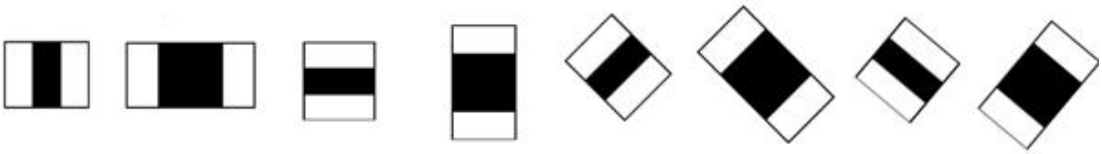


Figure 2.3 Line features of Viola & Jones method

Figure 2.4 Center and surround features of Viola & Jones method

Since the faces that are inside the images on which the face detection process is to be applied can come in different sizes, searched attributes need to be rescaled and searched within image again. Process of searching the whole image, for every single attribute, with scaled Haar attribute in different sizes can take a high computation time. Especially in applications planned to work in real-time, it becomes practically impossible. For this reason, in Viola and Jones method, Integral Image method is used for optimization of process.

Integral Image method is used to prevent summation, in every operation, of the pixel values of dark and bright areas in region Haar attribute is placed in stage of matching versions of each Haar attribute that are scaled according to different sizes, on image. In Integral Image stage, every cell inside the array is entered as summation of all figures over and left of it, according to formula stated below.

$$ii(x,y) = \sum_{\substack{x' \le x \\ y' \le y}} i(x',y') \tag{2.1}$$

On an image that is made into Integral Image format, summation of a particular area can be calculated with following formula, according to corner values.

$$I(x,y) = i(x,y) + I(x,y-1) + I(x-1,y) - I(x-1,y-1) \tag{2.2}$$

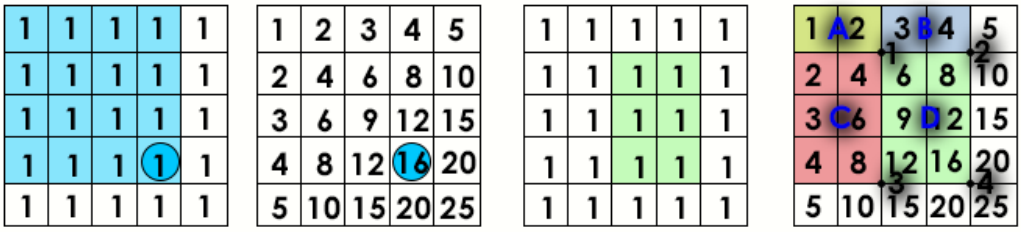



Figure 2.5 Integral image calculation phases

Because Haar attributes, that are to be used in the process of object detection in Viola and Jones method, can come in different dimensions and forms, a 24X24 resolution creates an attribute set over 160.000. This number is too high for any process of searching over image. While some of this attribute are relevant for detection of objects in image, some are not. For this reason, AdaBoost method is used to shrink this set by cleaning out the irrelevant ones among these attributes.

In AdaBoost method, each Haar attribute specified for face recognition process is regarded as a weak classifier. For each classifier, a particular weight value is described and a linear strong classifier function is structured for these classifiers. In this stage, each attribute's assigned weight value is changed with machine learning technique and error margin of function is reduced.

$$F(x) = \propto_1 f_1(x) + \propto_2 f_2(x) + \propto_3 f_3(x) + \cdots \tag{2.3}$$

While Viola and Jones method especially enables quite rapid detection of face images that are taken horizontally and from frontal perspective. Because it has a structure that depends on contrast of image, its success rate fluctuates in different

lighting angles. Image detection processes being feasible to run on devices with low computing capacities creates an advantage on terminal costs.

Comparissing of advanteges and disadvanteges of Viola and Jones algorithm is given in Table 2.1.

Table 2.1 Viola & Jones method advantages and disadvantages

| Advantages | Disadvantages |
|---|---|
| Detecting faces with different sizes | Different accuracy on different light positions |
| Detecting faces and face parth with high speed | |

### *2.1.2 Pixel Based Face Detection*

Similar to Viola and Jones' method of realising face detection process by using Haar attributes and AdaBoost training, face detection processes can be carried out over interrelations between selected pixels within the image.

Abramson, Steux and Ghorayeb suggested a method that allows carrying out the face detection process on pixel value differences of spots in image by making choices of attribute sets consisting of singular pixels on grayscale image (Abramson, Steux, & Ghorayeb, 2007). In suggested method, examining can be put into practice according to attributes composed of a particular number of pixels located in different positions within the image.

In this method, first step is to form attributes. In attribute forming stage, two different control sets are generated as $x_1, x_2, \ldots, x_n$ and $y_1, y_2, \ldots, y_m$ such that; $n$ and $m$ represents the number of pixels within the set, $K$ indicates the maximum number of control points to be selected for each attribute and $n, m < K$. While forming control set, a pixel selection is done in a way that $val(a)$ gives the value of pixel at point $a$ and for every $x$ and $y$ $val(x) > val(y)$.

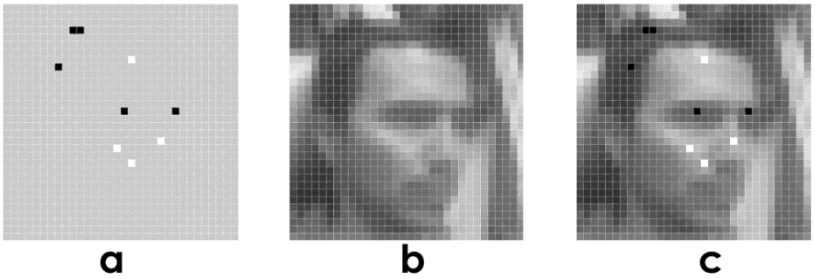



Figure 2.6 Sample feature for K = 5 (a), Sample grayscale image (b), Feature matching for image (c)

For every formed attribute, *val(x)* > *val(y)* should hold in all control points of attribute for selected window to procure the attribute. If this fails for any *x* and *y* value, chosen region is considered unsuccessful.

Abramson, Steux and Ghorayeb suggested a genetic-like method (Abramson, Steux, & Ghorayeb, 2007) as an alternative to AdaBoost for determining pixel based attributes which are to be used in face detection process and after initially creating 100 random attributes, they iteratively enhanced them as follows:

1. 90 attributes with highest error margins get eliminated
2. 50 new attributes are derived from 10 attributes with lowest error margins
3. 40 different random attributes are generated
4. This sequence of elimination and generating new attributes continue until progress halts

After attributes with highest success rates are selected, stage of searching for areas compatible with this attributes on image commences. Unlike Viola and Jones' method, Abramson, Steux and Ghorayeb scaled images with Image Pyramid method, instead of scaling attributes while searching with sliding window. In this method, searching is carried out, in versions of every image that are scaled in magnitudes of ½, ¼, ⅛, 1/16, along with the original.

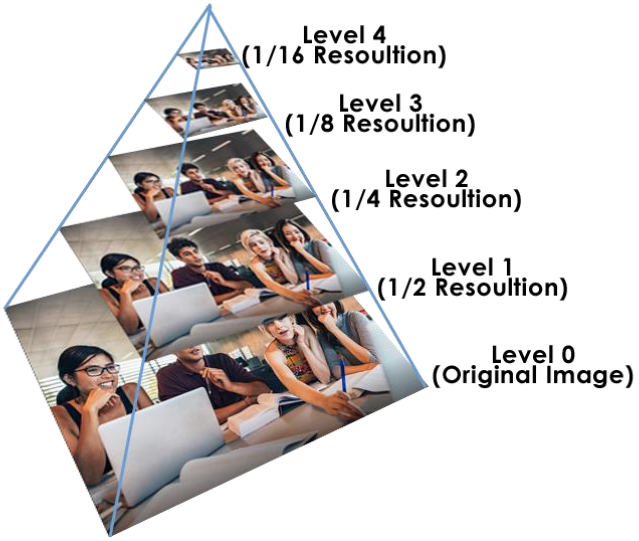



Figure 2.7 Image pyramid represantation with 4 levels

Abramson, Steux and Ghorayeb, with the method they developed, achieved high frame rates in computers with low processing capacity.

### *2.1.3 Face Detection With Skin Color In Color Images*

Another method used for face detection in an image is face detection with color analysis. Basically, face detection processes can be realised by using image processing methods on different color spaces like RGB, HIS and $YC_bC_r$.

Rein-Lien Hsu, Mohamed Abdel-Mottaleb and Anil K. Jain indicated that acquisition of face location can be achieved via color differences on images of eyes and mouth, which are the most basic distinctive features of a face (Hsu, Abdel-Mottaleb, & Jain, 2002). In this method, face detection can be done in any colored image by detecting and verifying eyes and mouth using color weights within the image. This method consists of two steps, which are; “Lighting compensation and skin tone detection” and “Localization of face features”.

In lighting compensation and skin tone detection stage, processes of brightness correction and detection of skin tone in image take place. Since skin tones can show variation under different lights, a lightning compensation is executed with “Reference

White" technique in this stage. After Reference White, colors that can be used for selecting skin tone are selected on lighting compensated image file.

While this operation can be actualised with different color spaces, Hsu, Abdel-Mottaleb and Jain $YC_bC_r$, which is one of the nine color spaces compared in the study of J. C. Terillon, M. N. Shirazi and H. Fukamachi, in their method. Furthermore, P. Kakumanı, S. Makrogiannis and N. Bourbakis achieved a true positive rate of 96.6% in their research using Hsu's method and $YC_bC_r$ space (Hsu et al., 2002).

In mentioned study, it is stated that within color space, high $C_b$ and low $C_r$ values around eyes and high $C_r$ values around mouth is found. To detect faces, these values should be examined and EyeMap and MouthMap need to be formed for areas where faces may exist. After the process of forming EyeMap and MouthMap, eye-mouth triangles should be detected and face estimation should be realised.

In more simplified methods, face detection procedure has been put into practice by forming only EyeMap and then carrying out face analysis processes on eye candidates on EyeMap. EyeMap building can be attained by creating two different maps as EyeMapC and EyeMapL and putting them through the process of bitwise and. Potential eye locations can be acquired using a specific threshold value over the built EyeMap.

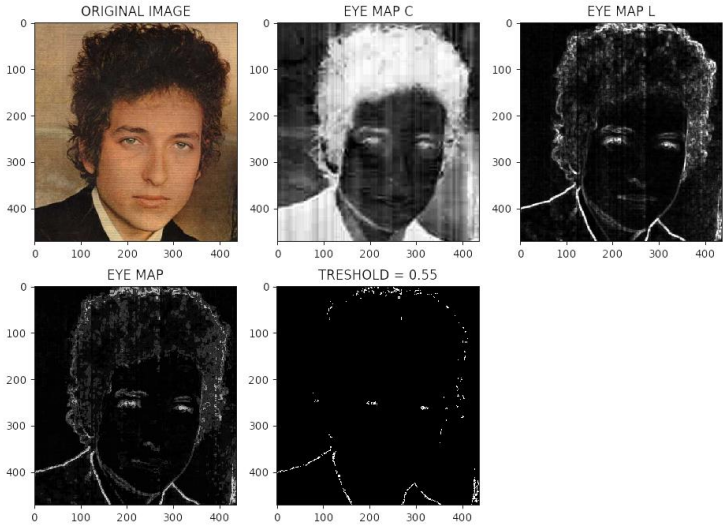



Figure 2.8 Original image, EyeMapC, EyeMapC, EyeMap and EyeMap with threshold t = 0.55

Procedure of detecting eye and mouth areas and detecting faces on colored images can vary based on features of image and nature of face inside the image. For example, in EyeMap forming process, horizontal angle of face or whether the person on image has glasses directly affects the detection success rate.

Table 2.2 Face detection with skin color advantages & disadvantages

| **Advantages** | **Disadvantages** |
|---|---|
| High accuracy on faces with different horizonral angles | High False – Positive rate |
| Hight True – Positive rate | Variable accuracy with different skin colors |
| | Low accuracy on non-frontal pictures |
| | Low accuracy on face detection for people who wears glasses |

### *2.1.4 Face Detection With Artificial Intelligence*

One method that is being utilised for face detection on image is Artificial Intelligence approach. Especially with Artificial Neural Network methods, object detection processes can be carried out in many ways. In this well studied area, H. A. Rowley, Shumeet Baluja and Takeo Kanade argued that face detection process can be realised with artificial neural networks, in 1997 (Rowley, Baluja, & Kanade, 1998a). In 1999, after that they studied that face can be detected with artificial neural networks, independent of its direction on image (Rowley, Baluja, & Kanade, 1998b). In time, capacity of processors has increased and "Graphics Processing Unit", which has the ability of executing parallel processes, started to be used. As a result, Deep Learning and Convolutional Neural Network approaches started to be utilised frequently in face detection and recognition operations.

If we examine Artificial Neural Networks at the basic, we can consider them as networks consisting of artificial neurons that act similar to biological neurons in our neural system and work on a mathematical function. Each artificial neuron takes one or more inputs and creates an output according to the formula that is defined for it.

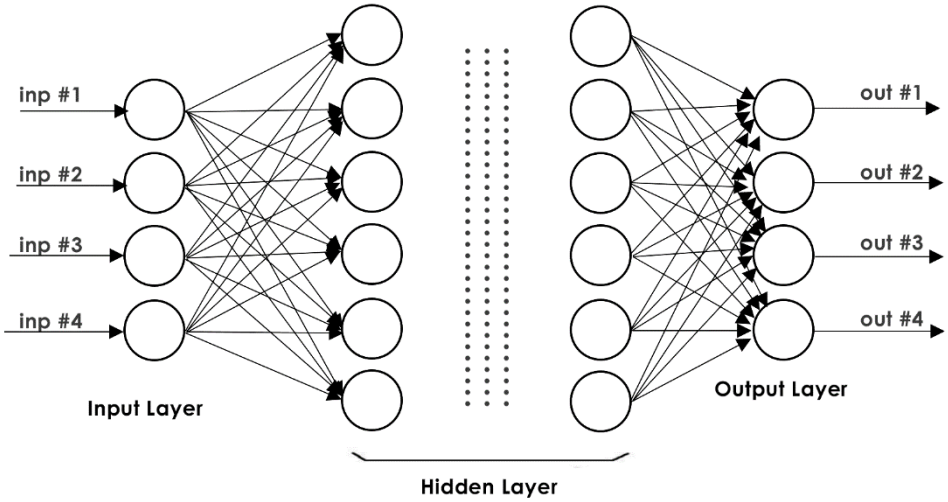



Figure 2.9 Artificial neurons and layers in artificial neural network

In artificial neural networks, artificial neurons exist in three different layers; “Input Layer”, “Hidden Layer” and “Output Layer”. While “Input Layer” and “Output Layer” consists of only one layer, “Hidden Layer” may have more than one. Although Input Layer artificial neurons in same number as number of inputs, Output Layer may have artificial neurons in different numbers dependent on nature of output. On the other hand, numerous layers within Hidden Layer can have artificial neurons in varying numbers.

Artificial neural networks can work in two ways, based on their structure. Structure, in which every neuron is transmitted only to following layer as input, is called “Feed-Forward” artificial neural network. In “Back-Propogation” artificial neural networks, however, each neuron not only feeds following layers, but also is linked to former layers and even its own layer as input.

Every output created by an artificial neuron, formed by one or more incoming inputs going through two functions with different tasks. While first of these is “Net Input”, which is made with linear combination of weight values dependent on inputs for each input, the second one is transfer function, which is applied before value generated by linear function is sent to next artificial neuron.

Depending on feeding system of artificial neural network, neurons in each layer is linked to by either previous neuron or both previous one and all following neurons as input. While artificial neural network is being constructed, for all neurons in each layer, random weights and bias values are initially determined. Net input function is formed based on these weight and bias values. Where the $w$ value is the input weight, the $a$ value is the input value and the $b$ value is the bias value specified for the function, the $F_{net}$ function for each artificial neuron is as follows.

$$F_{net} = (w_1 a_1 + w_2 a_2 + w_3 a_3 + \cdots + w_n a_n + b) \quad (2.4)$$

For each neuron, calculated net input value is transmitted to next level by going through transfer function. Transfer function that is being used in the stage of calculating output value of artificial neuron can have different structures. Transfer function can be specified as Threshold, Sigmoidal, Linear or Gaussian function. Output value is determined based on specified function type. Determined output value is transmitted to next layer.

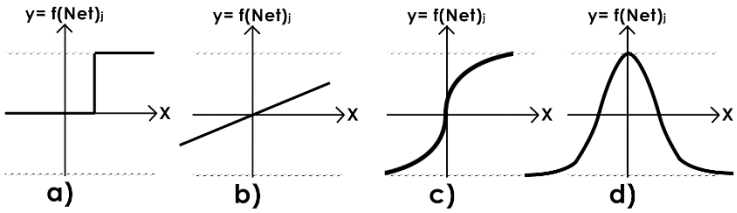



Figure 2.10 Transfer function types a) Threshold b) Linear c) Sigmoid d) Gaussian

While mentioned function structures are used as transfer functions, Rectified Linear Unit (ReLU) function is also often used in Deep Learning and Convolutional Neural Network structures.

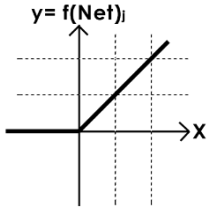



Figure 2.11 ReLU transfer function

$$f(x) = \begin{cases} 0 \; for \; x < 0 \\ x \; for \; x \geq 0 \end{cases} \tag{2.5}$$

Output of transfer function is identified as output of an artificial neuron and transferred to the artificial neuron where it will be utilised as an input value. In conclusion of this function executed in every neurons in every layers, output value of artificial neuron in Output Layer accepted as output value of artificial neural network.

In order for this formed artificial neural network to generate correct output values, it needs to be trained. In training of artificial neural network, aim is to correctly determine, for each artificial neuron, its designated bias value and weight values of outputs that is linked to it.

Artificial neural networks vary based on their learning structures. In "Supervised Learning" method, in which input values and corresponding output values are given, acquired outputs are compared to expected output values and training process is carried forward by making weight adjustments according to error or loss values.

In "Reinforcement Learning" method, result value of network's each iteration evaluated and process is carried out based on whether output is good or not. Network does a re-adjustment according to whether the output is good or not. Training process is realised with continuation of this process.

"Unsupervised Learning" consist only of entry of input information to network. No output is specified and there is no supporting activity for an output's success. Network forms a function structure for classification for each input.

In supervised learning, two different types of data with specified input and output values are formed for training process. One of these is “Training Data”, which will be used to determine weight and bias values in artificial neural network. Other one is the data which is outside of training data so training success of network can be measured.

Training process of artificial neural network can be defined as correctly determining weight and bias values of input values for each neuron that constitutes the artificial neural network. Before training of formed artificial neural network, weight and bias values of every neuron’s input values are randomly specified. Based on these initial values and by using calculated and expected output values of artificial neural network, a “Cost Function” is constructed. Approaches like “Quadratic Cost”, “Cross-Entrophy Cost”, “Exponentional Cost” or “Hellinger Distance” can be used for cost function (Nielsen, 2015).

Cost function is a function that takes, as input, all weight and bias values in artificial neural network and gives error value. All data in learning set are used as parameters for cost function. This cost function, which is built during the training process of artificial neural network with training data, is minimized.

While cost function is minimized during training process of artificial neural network, accuracy rate of network increases. However, in classification processes like object and face detection, success or network isn’t evaluated based solely on accuracy rate. Confusion matrix is used in calculation of network’s success rate.

| **ACTUAL** | | | |
|---|---|---|---|
| *Positive* | *Negative* | | |
| **True Positive** | **False Positive** | *Positive* | **PREDICTION** |
| **False Negative** | **True Negative** | *Negative* | |

Figure 2.12 Confusion matrix

Created confusion matrix gives the comparison between estimated and realised values. Based on these values, Recall and Precision values are, too, examined and network's succes rate is determined.

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives} \tag{2.6}$$

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \tag{2.7}$$

Artificial neural networks are used to detect particular objects on image. It is also used in detection and recognition of texts, objects and people in images. Artificial neural networks can be used by training it on every single pixel value of a digital image donnee. In general, pixel values of an image that is converted into grayscale format are given as input to artificial neural network.

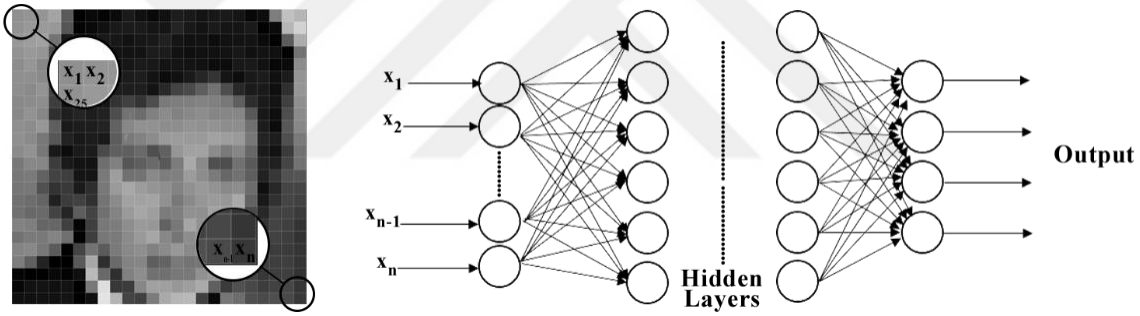



Figure 2.13 Giving input an image to artificial neural network

After training of artificial neural networks, classification process of image given as input is carried out and whether is belongs to a face is established. In order to detect the area in image where the face lies, with Sliding Window method, an area with specific size is taken from image and given to artificial neural networks as input; same are is slided and sweeping is actualised on the whole image. Since faces in image can vary in size, windows in different sizes can be used and/or in order to detect the area containing the face, image can be converted into different sizes by using Pyramid Representation.

In 1980, Kunihiko Fukushima has done a study about object recognition with artificial neural networks using Pattern Recognition (Fukushima, 1980). In his study, suggested a recognition method that is based geometrical similarity and isn't affected by object's position in image. Wisth this method, which will later develop and be accepted as Convolutional Neural Network, different object recognition studies has been practiced. In 1997, Steve Lawrence made a study on face recognition processes using Convolutional Neural Network (Lawrence, Giles, Tsoi, & Back, 1997).

In 2015, Haoxiang and Lin conducted a study for the face detection (Li, Lin, Shen, & Brandt, 2015). In their study, they used Convolutional Neural Network for the detection of faces that are directly from the side or up to a certain angle. They achieved a total recall value of 95.9% in the study using sliding window method. However, they have proposed the "Calibration Net" method to reduce the operating cost of the sliding window method. In their study, they achieved a recall value of 94.8% in the "12-calibration-net" method, which reduced the total number of processed windows to 1/13.

In 2015, Sachin Sudhakar Farfade, Mohammad Saberian and Li-Jia Li conducted a study on face detection in images with multiple faces, independent of direction (Farfade, Saberian, & Li, 2015). In this study, they reached over 90% recall values with a network containing 5 convolutional, 3 fully-connected layers. In the same year, Yi Sun, Ding Liang, Xiaogang Wang and Xiaoou Tang performed face and face recognition with the Convolutional Networks, where they used supervised training method in final and hidden layers (Sun, Wang, & Tang, 2015). They developed DeepID2 solutions that they have developed before and presented DeepID3 solution. With the solution they developed, they carried out studies on two different data sets. They obtained 81.4% accuracy in the public data set and 96% accuracy in the private data set.

Convolutional Neural Network consists of three different layers with different functionalities. Network is built by arrangement of these layers with different attributes and ordering.

#### *2.1.4.1 Convolution Layer*

Convolution Layer is the fundamental layer of CNN. This layer is comprised of attribute layers with sizes significantly smaller than the image. In this layer, whole image is put through kernel and dot product processes. After dot product, an activation array of image with smaller size than original is formed. There can be more than one Convolutional Layers in Convolutional Neural Network, with different sizes. Each Convolutional Layer and each kernel can execute process of detecting different attributes in image.

#### *2.1.4.2 Pooling Layer*

Pooling Layer, is used with aim of decreasing size and weights of array, formed as a result of activation process that is realized on activation layer which emerges as a result of Convolutional Layer, and making its attributes more distinct. Methods like Max Pooling, Min Pooling and Average Pooling can be used in this layer.

#### *2.1.4.3 Fully Connected Layer*

This layer, which takes the array resulting from Convolutional Layer and Pooling Layer processes as input, works as a fully connected neural network. Being in Standard Artificial Neural Network structure, this layer is used to classify output values in the last stage.

Success value of Convolutional Neural Network varies, depending on number, structure and order of utilized layers. Structure of a simple Convolutional Neural Network is given in Figure 2.14.

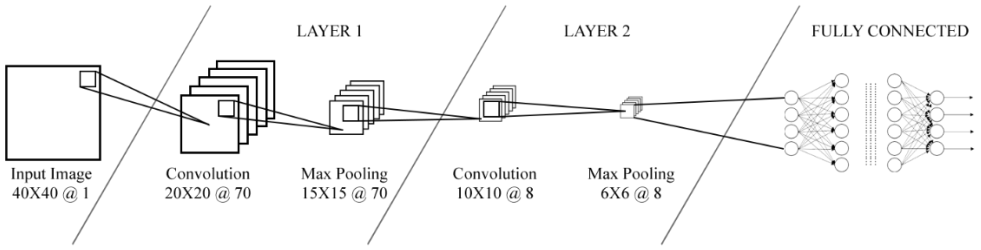



Figure 2.14 Sample convolutional neural network architecture with 2 convolutional layers

## 2.2 Face Recognition

Face recognition is the process of identifying a detected face and extracting some properties of it. This process is often carried out in conjunction with the face detection process. It can also be performed by adding certain classification functions to the face detection process. However, due to the discrete structure planned to be realized in the project, face detection operations and face recognition processes were separated from each other. Therefore, different facial recognition methods were examined.

Studies by Ekman and Friesen reveal that certain states of the facial muscles reflect the human mental state. However, their later "Facial Action Coding System" study has enabled the identification of certain facial facial expressions.

Ekman and Friesen's studies indicated that the mental state would reveal certain characteristics on the face. Based on this result, several different studies have been carried out to identify and classify features on the face. Basically, the facial recognition process consists of two stages: identification of facial features and classification of them. Different methods are used for each stage.

### *2.2.1 Feature Extraction*

Feature extraction is the stage of detecting and extracting facial features in order to classify the facial image according to certain classes. At this stage, appropriate and efficient characteristics of the classification are determined. Features resulting from

feature extraction are used in the classification stage. Many different feature extraction methods are proposed to be used at this stage.

*2.2.1.1 Histogram of Oriented Gradients (HOG)*

Histogram of Oriented Gradients is a feature descriptor used in feature extraction and image processing. It is a feature descriptor that is based mainly on the direction of gradient change in a specific area of the visual. In 2004, Navneet Dalal and Bill Triggs demonstrated human detection using the Histogram of Oriented Gradients with Support Vector Machine classification method (Dalal & Triggs, 2005). In 2011, O. Deniz, G. Bueno and J. Salido demonstrated the use of HOG for face recognition (Déniz, Bueno, Salido, & De La Torre, 2011).

In the HOG method, the first step is preprocessing. At this stage, resizing or color normalization is performed on the image. After this stage, pixel by pixed gradient vector calculation is performed in the image.

Gradient vector is the vector that shows the change of each cell in the x-axis and y-axis direction. If we accept f(x,y) function as a value function of at the point x and y, gradient vecter calculated as given below:

$$\nabla f(x,y) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} f(x+1,y) - f(x-1,y) \\ f(x,y+1) - f(x,y-1) \end{bmatrix} \tag{2.8}$$

$\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ values are the partial derivatives which give us the color changement on the x-axis and y-axis. Assuming $g$ is the vector size and $\theta$ is the vector direction, the formulas are as follows:

$$g = \sqrt{g_x^2 + g_y^2} \tag{2.9}$$

$$\theta = \tan^{-1}(\frac{g_y}{g_x}) \tag{2.10}$$

For example which is given in Figure 2.15, calculation of these values are given below.

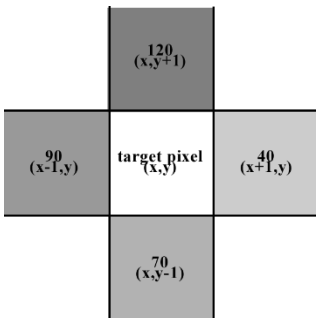



Figure 2.15 Sample point for calculate gradient vector

$$\nabla f(x,y) = \begin{bmatrix} 40-90 \\ 120-70 \end{bmatrix} = \begin{bmatrix} -50 \\ 50 \end{bmatrix} \tag{2.11}$$

$$g = \sqrt{50^2 + (-50)^2} = 70.7107 \tag{2.12}$$

$$\theta = \tan^{-1}\left(\frac{-50}{50}\right) = -45° \tag{2.13}$$

Gradient vector creation is performed as pixel-by-pixel in the whole image. As a result, two vectors are formed as size vector and direction vector. However, due to the high processing power requirement of this calculation process, these calculation operations can also be performed with different approaches. For this process, an array multiplication is generally performed with a particular kernel array. In their study, Navneet Dalal and Bill Triggs performed the Histogram of Oriented Gradients procedure using the kernels A= [-1,0,+1] and B=[+1,0,-1]. Accordingly $g_x$ and $g_y$ values are calculated as follows:

$$g_x = [-1,0,+1] * [f(x-1), f(x), f(x+1)] \tag{2.14}$$

$$g_y = [+1,0,-1] * \begin{bmatrix} f(x,y+1) \\ f(x,y) \\ f(x,y-1) \end{bmatrix} \tag{2.15}$$

Upon completion of the gradient vector creation, the image is divided into cells of $mxm$ size. For each cell, the distribution of the size and direction values is created. This dispersion is performed by dividing the selected direction range into specific parts. Distribution can be performed between 0 ° and 360 °, and it has been shown in the past studies that it gives more optimum results between 0 ° and 180 °. Dalal, in his study, divided the range from 0 ° -180 ° to 9 at 20 ° intervals. The distribution thus carried out gives the histogram of gradient values of the selected $mxm$ cell.

*2.2.1.2 Local Binary Pattern (LBP)*

The Local Binary Pattern method is a feature descriptor proposed in 1994 by T. Ojala, M. Pietikäinen, and D. Harwood (Ojala, Pietikäinen, & Harwood, 1994). It was based on the previously proposed Texture Spectrum model. In 2006, T Ahonen, A Hadid, M Pietikainen carried out a study on the use of LBP for face recognition (Ahonen, Hadid, & Pietikäinen, 2006). In 2016, Naoufel Werghi, Claudio Tortorici, Stefano Berretti and Alberto Del Bimbo used Local Binary Pattern as feature descriptor for face recognition (Werghi, Tortorici, Berretti, & Del Bimbo, 2016).

The basic LBP method works by dividing the image into 3X3 size arrays. It is based on the relationship of all cells with neighbors as pixel-by-pixel on the picture. Each pixel is considered the threshold value for the surrounding cells. Cells higher than the threshold value are considered as 1 and cells lower than the threshold value are considered as 0.

Once the Threshold matrix has been generated, each cell is written in binary format, clockwise and sequentially examined. The sum of the values written in binary gives the Local Binary Patern result for the calculated cell.

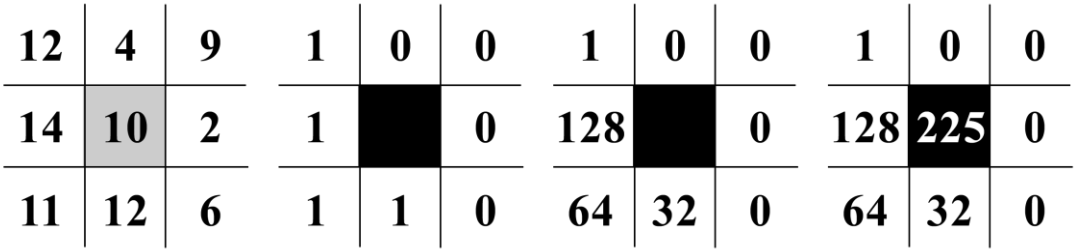



Figure 2.16 Sample LBP feature extraction of a pixel

Over time, LBP has continued to be developed through studies on different data sets. Instead of the basic LBP facing neighborhoods at a standard single unit distance, LBP methods for examining neighbors at 2 or more unit distances were also used. However, instead of taking the threshold value in the middle cell, different mathematical approaches are used, such as taking the mean value threshold in a 3X3 block. In Extended LBP, the threshold function is not used and is written to each cell by subtracting the value of the middle cell from the value of the cell. However, different patterns can be used for writing binary values in ELBP and multiple layers can be processed.

Zao and Pietikainen performed face recognition using the LBP method and Support Vector Machine classification. S Happy, Anjith George, and Aurobinda Routray performed face recognition in the same way (Happy, George, & Routray, 2012). S Jain, M Durgesh, T Ramesh used face recognition using LBP and Support Vector Machine - Artificial Neural Network approaches (Jain, Durgesh, & Ramesh, 2016).

*2.2.1.3 Gabor Filter*

Gabor filter is a filter used in image processing, pattern recognition and computer vision processes. It can also be used as feature descriptor. In 2014, E Owusu, Y Zhan and QR Mao performed face detection using the AdaBoost method with Gabor Features created using the Gabor Filter (Owusu, Zhan, & Mao, 2014).

Gabor filter is a filter formed by multiplying a harmonic function and Gaussian function. It is a linear filter. where λ is the cosine wavelength, ψ is the offset length, Θ

is the direction of the function, γ is the spatial viewing angle, the basic Gabor filter function is given below:

$$g(x, y;\ \lambda, \Theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \ \gamma^2 y'^2}{2\sigma^2}\right)\cos(2\pi\frac{x'}{\lambda} + \psi) \tag{2.16}$$

Functions of $x'$ ve $y'$ is given below:

$$x' = xcos\Theta + y\sin\Theta \tag{2.17}$$

$$y' = -xsin\Theta + y\cos\Theta \tag{2.18}$$

While Gabor filter basically creates a 3-dimensional function, 2-dimensional gabor filters can be used as kernel in feature extraction stage. Gabor Filter can be used for eye, iris and fingerprint recognition.

### *2.2.2 Feature Classification*

#### *2.2.2.1 Support Vector Machine*

Support Vector Machine is one of the methods used for classification. It is used in the classification of data belonging to more than one class. Support Vector Machine aims to find the most accurate hyperplane for classifying data in two-dimensional space, and to find the most accurate hyperplane for classifying data in higher-dimensional spaces.

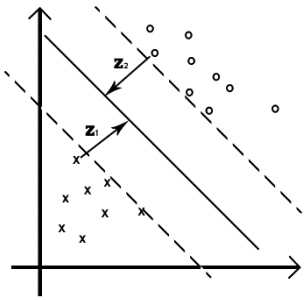



Figure 2.17 SVM binary classification

Of the data in each class, the points closest to this separator are called Support Vector. For the most accurate classification, the line to be selected must have the highest margin for these support vectors. Where Z1 and Z2 values in Figure 2.17 give the distance between choosen hyperplane and support vectors and $g(\vec{x}) = \overrightarrow{\omega^T}\,\vec{x} + \omega_0$ is the vector function; the function which gives the margin of the hyperplane is given below .

$$Z = \frac{|g(\vec{x})|}{||\omega||} = \frac{1}{||\omega||} \tag{2.19}$$

$$\frac{1}{||\omega||} + \frac{1}{||\omega||} = \frac{2}{||\omega||} \tag{2.20}$$

The value of $\vec{\omega}$ have to be minimized to increase the margin of the hyperplane to be selected. This process is a non-linear optimization process and different methods can be used in the minimization process. The basic Support Vector approach has been developed for binary classification operations. As the number of classes increases, the success rate of the basic Support Vector Machine approach decreases. Support Vector Machine approach to higher level classed data, Support Vector Machine approach is extended to One-Against-All and One-Against-One approaches.

Many different studies have been carried out on emotion recognition using Support Vector Machine. After the face detection process is performed with different methods, it is used in the classification process through the features extracted from the picture. For example, the basic Support Vector Machine approach can be used to distinguish between happy and unhappy human characteristics on two different feature datasets formed from images of happy and unhappy people. E.M.Bouhabba, A.A.Shafie and R.Akmeliawati performed facial detection using Haar-Like Features and performed emotion detection using the Support Vector Machine technique (Bouhabba, Shafie, & Akmeliawati, 2011).

*2.2.2.2 ID3 Decision Tree*

The ID3 algorithm introduced by Ross Quinlan in 1986 is a machine learning algorithm for building a decision support tree (Quinlan, 1986). It is mainly based on

entropy calculations. It starts by first calculating the entropy of the entire data set. After the general entropy calculation, it calculates the entropies of all attributes on the given training data set. Calculates the total gain for each attribute based on the overall entropy value and the entropy value of each attribute. Performs branching according to the attribute with the highest gain value.

Where p is the number of data included in each class and H is the general entropy value, general entropy value function in a data which has N different classes is given below.

$$H(p_1, p_2, \dots, p_n) = \sum (p_i \log(1/p_i)) \tag{2.21}$$

After the general entropy calculation is performed, the entropy calculation for each attribute is performed with the formula given above too. For each attribute, where G is the gain function, gain is calculated with the function given below.

$$G = H(p_1, p_2, \dots, p_n) - \sum P(D)H(D) \tag{2.22}$$

After calculating the gain value for each attribute value, branching is performed according to the attribute with the highest gain value. If there is no root value, the first highest calculated value indicates the root of the tree.

Ross Quinlan developed C4.5 and C5.0 algorithms by developing the ID3 algorithm. In addition, different approaches to this algorithm have been developed. N. Sebe, M.S. Lew, Y.Sun and I. Cohen performed a mental state analysis on the extracted images with the ID3 algorithm (Sebe et al., 2004). They performed their studies on the Authentic expression database and Cohn-Kanade database with a 95% confidence interval. In their study on Authentic Expression Database, they achieved classification error rate between 6.96 - 9.76% with ID3 algorithm. In their study on the Cohn-Kanade Database, they achieved an error rate between 10.70 and 16.70%.

*2.2.2.3 K-Nearest Neighbors*

In 2017, M. Nazir, Z. Jan and M. Sajjad conducted a study on emotion recognition using the Histogram of Oriented Gradient method (Nazir, Jan, & Sajjad, 2018). They used the K-Nearest Neighbor method in classification phase as classifier after feature extraction. They used MMI and CK + datasets in their studies.

K-Nearest Neighbor is a classification method that can be trained with supervised and unsupervised learning methods. K in the definition of K-Nearest defines the number of nearest neighbors to be used in the method. Each object to be classified is defined according to the class of the nearest K number in the training data set. Each element is defined as the class of the nearest K number of elements belonging to that class. Euclid Distance is usually used to select the nearest neighbors.

# CHAPTER THREE
# METHODS

In this part of the thesis, the method followed about the emotion detection process from the visuals is explained. The methods used in carrying out the thesis subject are examined in detail in the "Related Works" section. In this section, the overall structure will be explained as a whole. Each piece will be examined in general architecture.

The emotion recognition from image sources process was performed in three stages. GDPR and KVKK (Protection of personal data law in Turkey) have been taken into consideration while carrying out the holistic design of the phases. The process is carried out in three different stages which are carried out in two different environments. The first of these environments is the environment where the face is detected on the visual and feature extraction is performed. The output from this section is feature data that does not contain any personal data. The second environment in the application architecture is the environment where emotion recognition process takes place via face features. The figure explaining the general architecture of the system is given below.

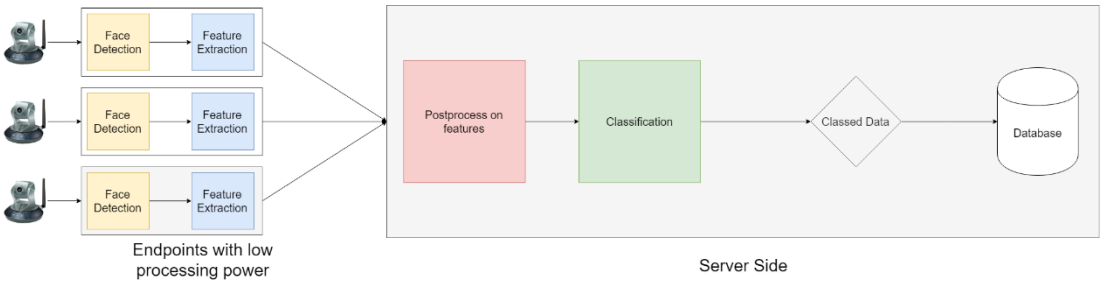



Figure 3.1 Overall system architecture

In the visually outlined design, the images obtained from the cameras are preprocessed at low cost terminals. In this section, only face detection and feature extraction are performed in the most efficient manner in order to reduce system costs. The output from this section is transferred to the server side section, which has a higher

processing power. In this section with high processing power, final processing is performed on visual data. The data obtained as a result of the postprocessing process is entered into the classification process. The classification result obtained from the visual data is recorded as anonymous on the database.

## 3.1 Low Processing Power Endpoints

Low Processing Power Endpoints (LPPEs) are the first part of this application. In this part, one LPPE is connected to one or more visual data source for data collection. Because of the limitation of each LPPE, there may be more than one LPPE for each point where the data is gathered. Therefore, the number of LPPE that may be needed for each area of use increases. Because of the potantial number LPPEs needs, LPPE costs need to be reduced. In order to reduce LPPE costs, mini-computers such as Raspberry Pi were preferred in this section.

In this LPPE unit, face detection and feature extraction processes is made. Each LPPE has Linux a based operation system. They connect to visual data source via Ethernet or USB. LPPEs with ethernet connectivity are able to read visual data from Ip Camera data sources. USB type LPPEs read data directly from camera through USB connection. Each type of LPPEs read data as continously as they are able to process. For each image that they read is processed immediately and send to server side. They store processed data into an internal storage while they don't have any connection to the server.

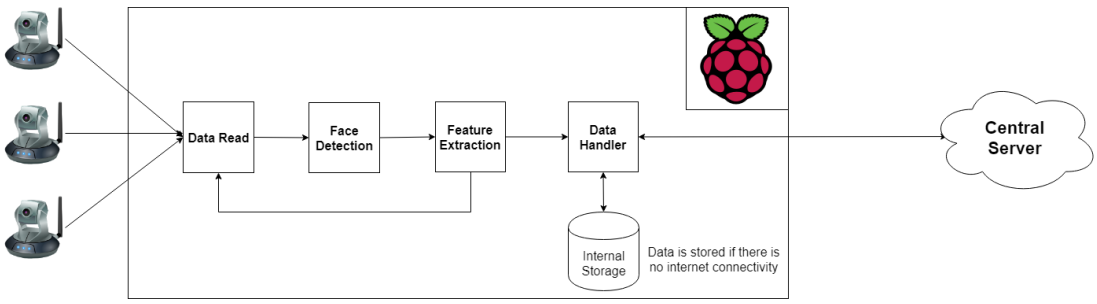



Figure 3.2 Overall system architecture

### *3.1.1 Face Detection*

The study about face detection aproaches is caried out in detailed in related work part. While choosing the best face detection method and library, two criteria was evaluated in same time: accuracy and performance. Most commonly used face detection open sourced libraries OpenCV and DLib both have face detection functionalities with several face detection tecniques. The accuracy and performance comparission of both libraries is given in Figure 3.3.

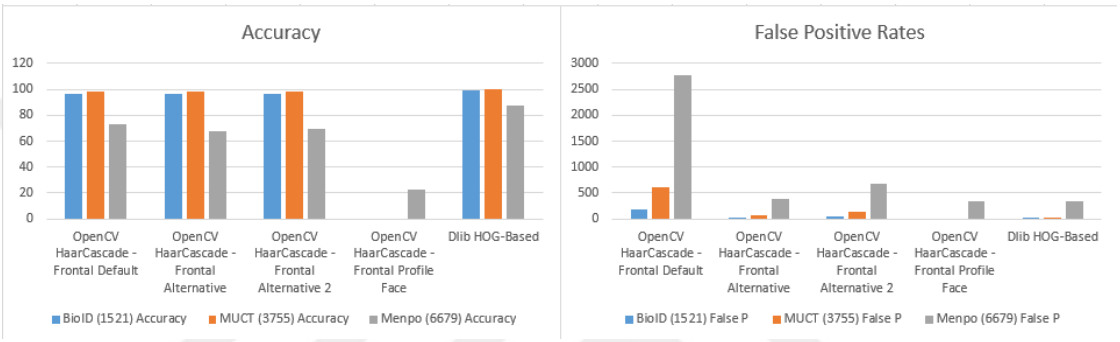



Figure 3.3 Face detection libraries comparision

In addition to accuracy rates of these aproaches, performence tests were ran in a sample LPPE whose hardware specification is given below:

- 1.5 GHz 32-bit-quad-core ARM Cortex CPU
- 2 GB LPDDR4-2400 Ram
- MicroSDHC Storage Disk

As the result of the performance test, OpenCV Haar-Like Based library detected faces with 6 frames/second rate. Dlib – Neural Network Based library detected faces with 4 frames/second rate and Dlib – HOG-Based face detection library detected faces with 8 frames/second rate.

Altough there are too many approaches like Haar-Like Based, Convolutional Neural Network Based and Neural Network Based face detection aproaches, HOG-

Based face detection Dlib library was selected for face detection process in LPPE. because of accuracy and resource usage.

Each detected face is extracted from the image. In this step there is not any preprocess on the extracted face image. To increase feature extraction success rate from face image, the image is not resized. Extracted data is transfered to feature extraction step with their orginal sizes.

### *3.1.2 Feature Extraction*

In feature extraction phase, Dlib's Facial Landmar Detector library is selected for feature extraction process. Dlib's Facial Landmark Detector library extracts 68 facial landmarks from given image which are oftenly used for classify action units on a face. Each specific intervals of the extracted 68 face landmark values identify specific areas on the face. Landmark point ranges for each face section is given in the table 3.1 and illustration on facial landmarks is given in Figure 3.4.

Table 3.1 Landmark intervals for each face shape

| Face part | Interval |
|---|---|
| Jaw | 0-17 |
| Nose Bridge | 27-30 |
| Nose Bottom | 31-35 |
| Left Eye | 42-48 |
| Right Eye | 36-42 |
| Left Eyebrow | 22-27 |
| Right Eyebrow | 17-22 |
| Inner Mouth | 60-67 |
| Outer Mouth | 48-59 |

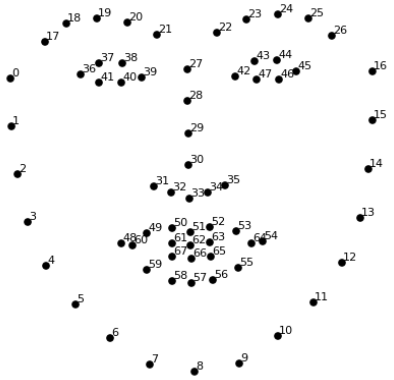


Figure 3.4 Facial landmarks

In feature extraction phase, 68 facial landmark points are extracted for each detected face in a image. These facial landmark points form an array with 68 bytes size. LPPEs transfer this facial landmark points array to classification server while they have internet connection continously with LPPE's identifier. They store these data on local disk when they don't have any connection with classification server till they have internet connection. When the internet connection restored they send accumulated data to classification server.

The data which is transfared between LPPEs and classification server doesn't have any identifier information about the detected personal. They are only facial features for classification and nobody can match them with a specific person unless they have any facial features of the people.

## 3.2 Classification Server

Classification server is the second part of the system architecture. It's responsible for collecting data and classify them for emotion recognition. Main function of classification server is classifying facial landmarks to emotions. It has two seperated databases. One database is used for storing raw facial landmark data and another one is used as relational database. Architectural design and primary technologies are given in figure 3.5.

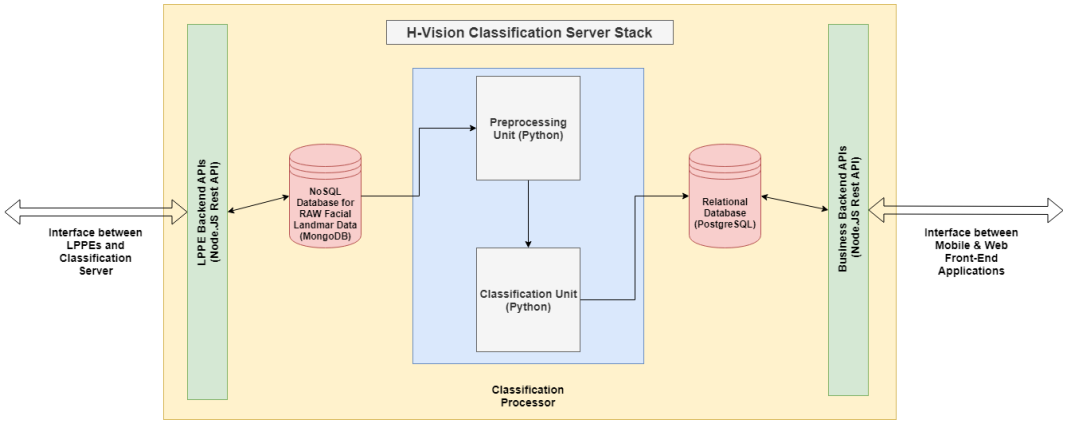



Figure 3.5 Classification server architecture

### *3.2.1 Interface APIs*

Both LPPE Backend APIs and Business Backend APIs were placed as communication interfaces both with LPPEs and Mobile or Web front-end applications. Both were developed on Node.JS with using Express.JS library and they are using OAuth 2.0 as authorization protocol. LPPE backend API is responsible for collecting raw data from LPPEs and store the data to non-relational NoSQL database. MongoSQL is selected for storing non-relational data. Also, LPPE backend is responsible for track LPPE end point status and provide them system updates.

Business Backend APIs is responsible for providing backend services to front-end applications. It uses PostgreSQL as the database provider. It provides emotion tracking reports for each individual LPPE or group analysis report for predefined LPPE groups. It provides several web services for serving analysis these reports, authentication proccesses, configurating system configurations.

Both LPPE Backend APIs and Business Backend APIs were isolated from classification process. They are responsible for only data storage and run analysis process on classified data.

### *3.2.2 Classification Processor*

Classification Processor is the main part of the business. It's responsible for preprocess facial landmarks and classify the data. It consist from 2 seperated part. One part is responsible for preprocessing part for facial landmark data and another one is responsible for classifying these data to emotion classes. These parts are fully seperated from API interfaces. They turn raw data to classified data with reading them from NoSQL database and writting into relational database.

#### *3.2.2.1 Preprocessing Unit*

First process which runs in the preprocessing unit is the face alignment. With this process, gathered facial landmarks is aligned for increasing accuracy rate in the classification process. For this process FaceAligner feature of DLib library is used.

Second process which is following the face alignment process is face illustration process. Because of the privacy causes limitation, system stores only facial landmark data. For this reason, Convolutional Neural Network approach doesn't aplicable on raw data. Illustration process gives the system allowment to run Convolutional Neural Network based emotion recognition. In last step, preprocessing unit convert alligned landmark data to face illustrations and send them into classification unit.

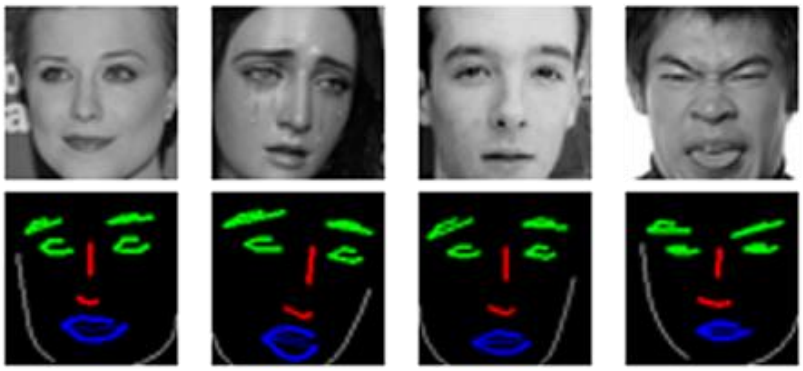


Figure 3.6 Face illustrations of detected faces

*3.2.2.2 Classification Unit*

Classification unit is the part which is responsible for emotion recognition from processed data. Emotion recognition process is caried out with artificial intelligence in this part. Convolutional Neural Network approach is selected as artificial intelligence solution in this part. For implementing Convolutional Neural Network in this part, Tensorflow library was used. Designed and trained Convolutional Neural Network model is described in details in Application section with its results.

# CHAPTER FOUR
# APPLICATION

The developed application consists of many different parts. Each piece is responsible for different tasks. During the development, development processes were performed on different platforms for each part. When selecting platforms, ease of development and performance criteria were taken into consideration. In Classification Server, APIs have been developed on the node.js platform. Python has been chosen as the development platform in "Preprocessing Unit" and "Classification Unit" where artificial intelligence and image processing are performed. NoSQL was preferred for the storage of raw data and PostgreSQL was preferred as the relational database because of the need structures were different.

## 4.1 API Development

API development is provided by using Visual Studio Code application. RESTFull web services have been developed by using the Express.JS library on the Node.JS platform.

Node.JS is a Javascript Runtime that was developed in 2009. Runs the software developed on the Javascript language by compiling it into the machine language. It enables the realization of fast and scalable server software. It can work with many libraries externally and supports asynchronous operation. It is a platform that provides high performance and low resource usage. Many companies like LinkedIn, Trello, Paypal and Uber use the Node.JS platform. LinkedIn has moved its Java-based software to Node.JS, reducing the resource requirement of the current software system to 1/10 in the past.

Express.JS library is used to create RESTFull web services on Node.JS. Express.JS is an easy-to-develop and flexible framework that runs on node.js. The services of the application have been improved with Express.JS installed as a package to Node.JS. POST and GET services were used in the development process. The Express.JS library

allows middleware in the service response process. Thus, before or after the service workflow, it provides logging, authorization and similar processes.

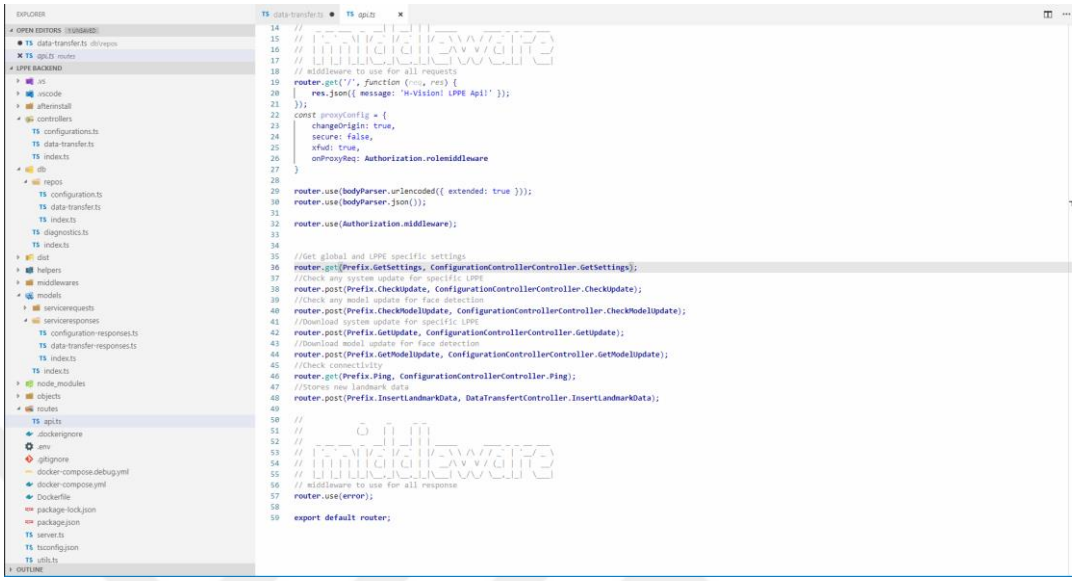


Figure 4.1 LPPE backend API application

The application is basically composed of 3 parts. The first section is the "Controller" section that performs the business workflow in the system. Workflows to be performed according to the requests received on this section are carried out. The second section is the "Repositories" section where database operations are performed. This section performs CRUD operations on the database by performing database connection operations. In the LPPE Backend API, the repository section uses the "mongodb" library of Node.JS. It executes the operations it will execute with Mongo database through this library. In the mobile and web interface services, "pg-promise" library was used to interact with the PostgreSQL database.

The last part of the system is the "routers" section where the service routing is performed. Here, a controller function assignment for each http or https address is defined in the system. A different controller can be assigned to each URL according to the GET or POST function. Also in this section, middleware assignments for service operations can be performed. This feature of Express.JS allows the development of asynchronous and controllable web services.

Once the Node.JS platform is developed, it is compiled and run. Node.JS platform, which can be developed with Typescript and Javascript software development languages, generates Javascript code as a result of compilation process. Node.JS is platform independent, and this compiled code can be run on platforms with the Node.JS environment. The developed application can run in Windows, Linux and iOS environment. The API sections of the application run on servers with Debian operating system created on Google Cloud.

### *4.1.1 MongoDB*

Within the scope of the project, MongoDB, a NoSQL solution, was used to store the data collected over LPPEs. As a database server, the Compute Engine running on Google Cloud was created. The system features created are 50GB SSD disk, 2 Virtual CPUs and 8GB Ram. In addition, a 100 GB physical HDD is used for data storage.

MongoDB consists of storages called Collections. Within the scope of the project, a collection was created in order to collect unprocessed facial landmark data. The data in this collection holds the LPPE id from which the data was collected, the 68 landmark data collected, and the date the collection was performed. Adding records to this database is carried out through the Node.JS web services on the LPPE side.

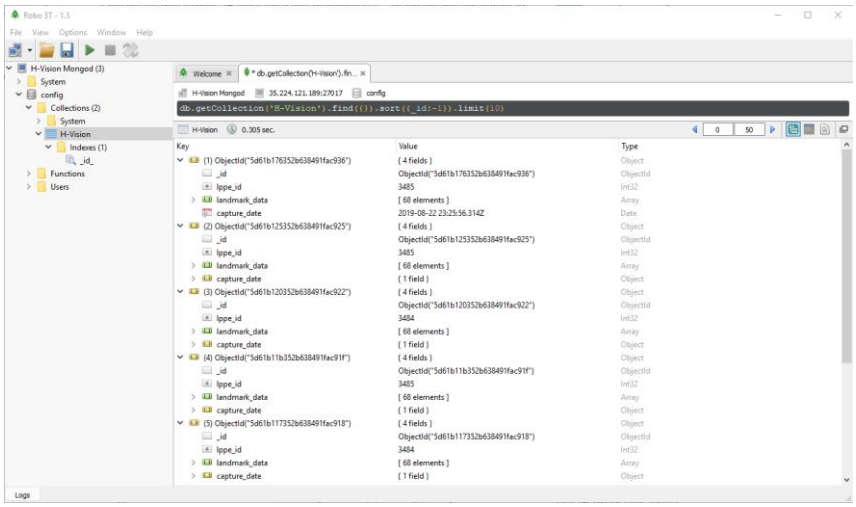


Figure 4.2 MongoDB landmarks collection

The MongoDB database stores only non-relational raw data. The LPPE id in this data connects the source to which the data is collected and the relational database. The unique id value of each data recorded on MongoDB is stored in the relational database with the class data resulting from data processing. This ensures that the class registered in the relational database matches the raw data, but there is no connection between the two databases.


Figure 4.3 Node.JS MongoDB facial landmark insert implementation

### *4.1.2 Relational Database*

In the developed system, relational database is used for keeping face landmark classifications and performing frontend operations. PostgreSQL database is preferred as relational database solution. PostgreSQL is an open-source database solution. Developed and supported by the active community. PL / SQL as well as Pythoni Perl, C ++ and R database development operations can be performed. Replication servers can be created with Streaming Replication and Slony-I replication methods.

PostgreSQL contains user information stored in the system, LPPE terminal information of users, LPPE groups created by the user, and classification results of data collected from LPPEs. The results of the classification process are recorded in the lppe_data table. This table contains the classification result of each data read from LPPEs. A separate record is created for each face in each data read from each LPPE. In this table, 7 basic emotion estimation rates are kept for each face data. In addition, the emotion estimated at the highest rate for each face data is kept.

In order to allow the system to work with more than one LPPE in certain regions, there are groups of LPPEs containing more than one LPPE. The relational database contains the lppe_group table to group multiple LPPEs. This ensures that the LPPE belonging to any lppe_group can be reported individually or as a group. In this way, large departments can be reported as a group at the points that use the system. The ER Diagram of the relational database is given in Figure 4.4.

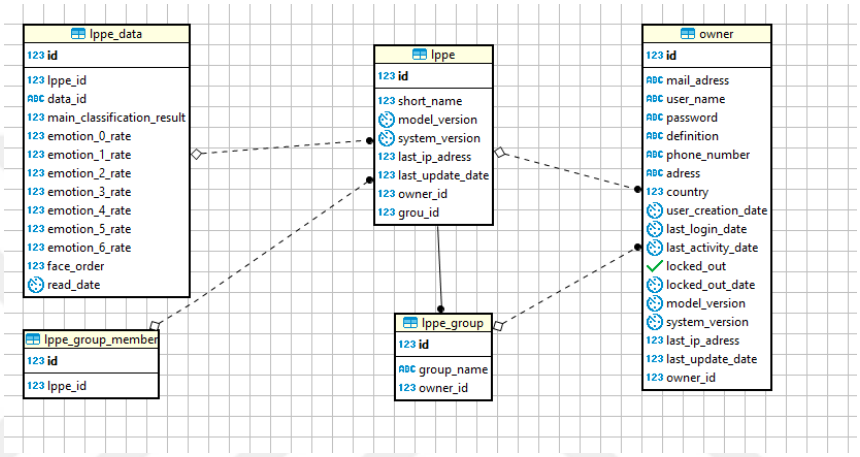



Figure 4.4 ER diagram of relational database

## 4.2 Classification Processor

In the classification process section of the application, preprocess process, illustration process and classification process are performed on face data. All improvements made at this stage have been made with Python. Python 3 is used as the Python programming language version. OpenCV and DLib libraries were preferred for preprocess and illustration processes, and Keras and Tensorflow libraries were used for classification.

The Classification process division runs on a separate Compute Engine on Google Cloud, separate from other system components. The system works as a black box and receives raw data from MongoDB and saves the output to the PostgreSQL database without any other workflow.

### *4.2.1 Preprocessing Unit*

Preprocessing Unit is the section where alignment and illustration of raw data collected from LPPEs are performed. Developed with Python programming language. Performs face alignment on the face data received primarily. When performing this operation, the midpoint calculation is performed for both eyes. The angular value between the line passing through the midpoint and the x-cordinate is calculated. According to the calculated angular value, the array containing facial landmark data is rotated with the Numpy library. Numpy library is a library that provides mathematical functions on Python.

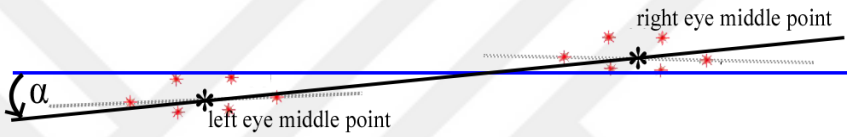



Figure 4.5 Face alignment

After rotation, the data is parsed for each face piece. For each face part, polyline drawing is performed using the OpenCV library. At this stage, optimization processes were performed according to the training results in the classification process. In order to increase the success rate in the classification process, each face part is drawn in different colors and different line sizes. The color and line dimensions selected for each face part in the drawing process are given in Table 4.1.

Table 4.1 Face part colors and line weight for face illustration

| Face part | Red | Green | Blue | Line Weight |
|---|---|---|---|---|
| Jaw | 255 | 255 | 255 | 1 |
| Nose Bridge | 0 | 0 | 255 | 2 |
| Nose Bottom | 0 | 0 | 255 | 2 |
| Left Eye | 0 | 255 | 0 | 2 |
| Right Eye | 0 | 255 | 0 | 2 |
| Left Eyebrow | 0 | 255 | 0 | 2 |
| Right Eyebrow | 0 | 255 | 0 | 2 |

Table 4.1 continues

| | | | | |
|---|---|---|---|---|
| Inner Mouth | 255 | 0 | 0 | 1 |
| Outer Mouth | 255 | 0 | 0 | 2 |

After the rotation and drawing operations are performed, the image created is temporarily saved to disk. The Classification Unit processes each image saved on the disk and performs the classification. The recorded picture is deleted from the disc after classification.

### *4.2.2 Classification Unit*

Classification unit is the last point where the classification process takes place. In this section, the classification process is performed on the images created by the preprocessing unit. Classification is carried out by Convolutional Neural Network approach. Python programming language and Keras and Tenserflow libraries are used for Convolutional Neural Network implememntation. The model training process was runed on Compute Engine which has pre-installed Tensorflow system on the Google Cloud platform. It has 4-Virtual CPU, 16GB Memory and a NVIDIA Tesla K80 TPU which enables training with TPU.

Tenserflow is an open source artificial intelligence library developed by Google, announced in 2015, which is widely used in the field of deep learning. It can be used with many different programming languages. It enables model trainings on developed artificial intelligence models with one or more CPUs or GPUs. Particularly with GPU, it provides great advantages in performance of model trainings. Keras is a high-level API that runs on the Tensorflow library. It can be developed on Tensorflow and provides easy coding of complex models. It is written in Python programming language. These two libraries were used in the creation and training of the Convolutional Neural Network model.

For the model training, firstly the determination and processing of dataset were performed. FER-2013 dataset was used for emotion detection model training. The FER-2013 data set was developed by Pierre-Luc Carrier and Aaron Courville. It was

first used in the deep learning competition held in Kaggle. It consists of 35,887 pictures containing 7 different emotions. All images in the data set are labeled and each image has 1 emotion label. All images are 48x48 pixels in size and grayscale.

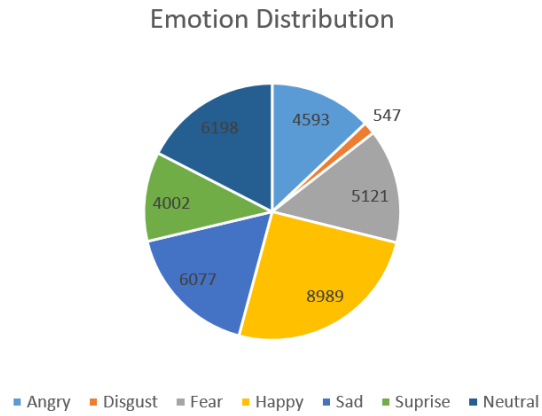



Figure 4.6 Emotion distribution of FER-2013 dataset

State-of-the-arts value was 75.1% in previous studies with Fer-2013 dataset (Zhang, Luo, Loy, & Tang, 2015). In human studies conducted on the same data set, the accuracy value was measured as 65 ± 5%. In the Kaggle Competition held in 2013, the first 3 ranks were 71.16%, 69.26% and 68.82% (Goodfellow et al., 2015).

In the study, first of all, model creation and training operations were performed without any pre-processing process over raw data set and the results were evaluated. Different Convolutional Neural Network architectures have been established. In the study; 2, 3 and 4 convolutional layer models were studied. Each structure was measured with different feature numbers, pooling values and activation functions.

In the selected model, 2x2 size features were preferred for convolutional function. In the first Convolutional Neural Network layer, a single convolution function with 64 features was chosen. The number of features was doubled in each of the next 3 layers, and 3 convolution functions are preferred in each layer. After each convolution function, Batch Normalization and RELU activation was performed. 50% dropout was performed at each convolutional layer to prevent overfitting. In the last stage, 2 Fully Connected layers were used by Flatten process. After each fully connected layer, Batch

Normalization and RELU activation function were applied. 35% dropout was performed after each RELU activation to avoid overfitting in the fully connected layer. Finally, a classification was made with a dense layer that determines the emotion label. The architecture is given in Figure 4.7.

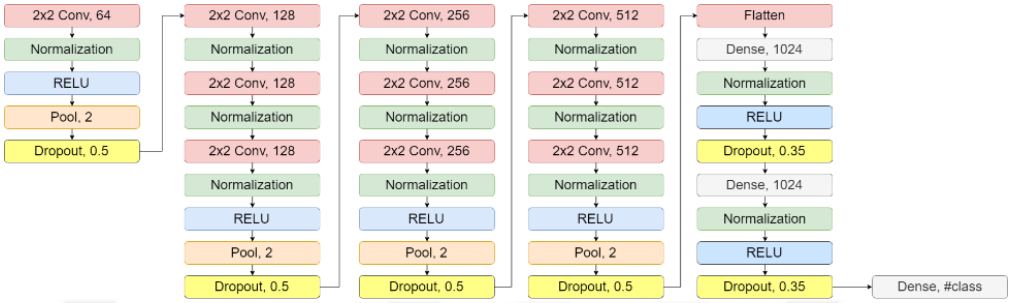



Figure 4.7 Builded Convolutional Neural Network model

After Convolutional Neural Network model had been decided, the data was separated as training and validation data. %25 data has been selected randomly as validation data from whole image dataset. Training process has been accomplished on the raw data with 200 Epoch. The training with the NVIDIA K80 in the Compute Engine on Google Cloud took about 210 minutes. After training process validation process was ran over the validation data. Validation data have 8971 images with all labels. Over these 8971 images, 5525 correct classification were made with the builded architecture. The accuracy was measured as 61.58% on the raw validation data. Recall, precision and prediction result is given in the Table 4.2.

Table 4.2 Original FER-2013 model results for each image label

| Emotion | Angry | Disgust | Fear | Happy | Neutral | Sad | Suprise |
|---|---|---|---|---|---|---|---|
| False Negative Rate | 52.51 | 44.61 | 51.78 | 17.82 | 52.26 | 23.26 | 41.08 |
| False Positive Rate | 06.67 | 00.79 | 07.57 | 06.19 | 10.32 | 03.44 | 11.01 |
| True Negative Rate | 93.32 | 99.20 | 92.42 | 93.80 | 89.67 | 96.55 | 88.98 |
| True Positive Rate | 47.48 | 55.38 | 48.21 | 82.17 | 47.73 | 76.73 | 58.91 |

Table 4.2 continues

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Precision | 53.60 | 50.70 | 52.22 | 81.40 | 48.18 | 73.55 | 52.71 |
| Recall | 47.48 | 55.38 | 48.21 | 82.17 | 47.73 | 76.73 | 58.91 |
| Accuracy | 86.92 | 98.57 | 85.94 | 90.91 | 82.65 | 94.34 | 83.79 |

After model training with raw image data model training process was performed with processed visuals. Each picture in the FER-2013 data set was processed in the LPPE and Preprocessing Unit. At the face detection process on the LPPE unit face detection accomplished on 17171 images. After that, each images processed on the Preprocessing Unit. Face illustrations were created over the face landmark values that were alignmented. After this processes 25% of data were selected as validation data and rest of the data were selected as the training data. Using the same model, model training was carried out in the same way. As a result of 200 epoch, 72.85% training set accuracy and 56.21% validation data set were obtained. Recall, precision and prediction result for processed data is given in the Table 4.3.

Table 4.3 Processed FER-2013 model results for each image label

| Emotion | Angry | Disgust | Fear | Happy | Neutral | Sad | Suprise |
|---|---|---|---|---|---|---|---|
| False Negative Rate | 64.12 | 58.10 | 70.71 | 18.86 | 71.65 | 30.26 | 39.78 |
| False Positive Rate | 05.73 | 01.80 | 06.22 | 10.16 | 08.06 | 03.57 | 17.62 |
| True Negative Rate | 94.26 | 98.19 | 93.77 | 89.83 | 91.93 | 96.42 | 82.37 |
| True Positive Rate | 35.87 | 41.89 | 29.28 | 81.13 | 28.34 | 69.73 | 60.21 |
| Precision | 48.55 | 28.97 | 38.38 | 77.07 | 34.56 | 71.48 | 45.09 |
| Recall | 35.87 | 41.89 | 29.28 | 81.13 | 28.34 | 69.73 | 60.21 |
| Accuracy | 86.60 | 97.22 | 86.23 | 87.25 | 83.62 | 93.38 | 78.08 |

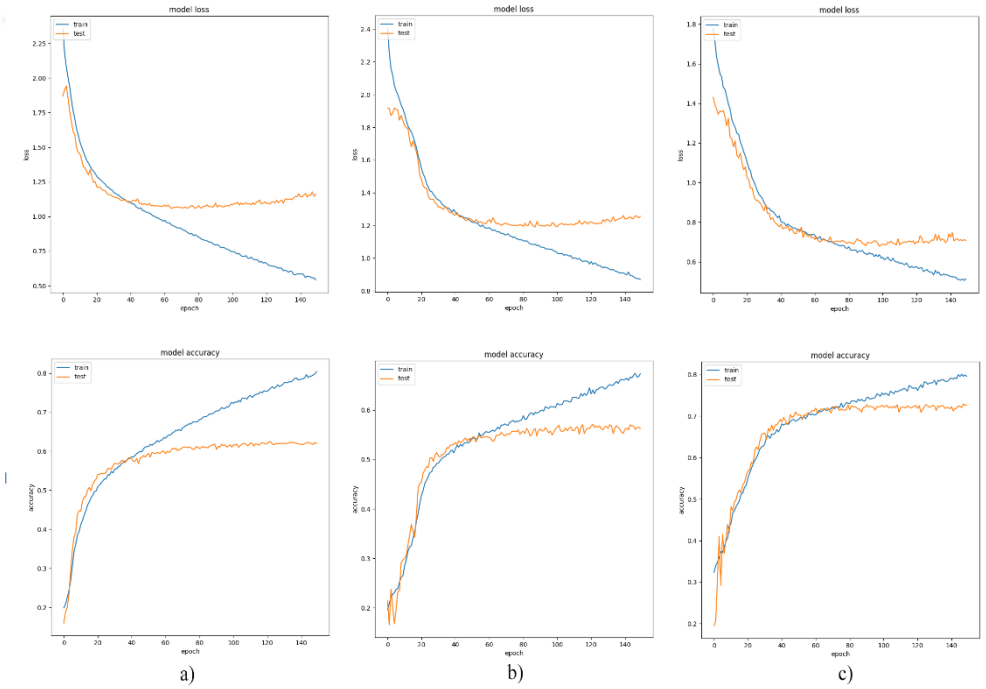
Finally, angry, happy, natural and sad emotion data were separated on the data set and training was performed on these data. Basic emotion data set contains 11483 images and 2870 images were selected as training data set. In the training with the

same model, training was performed with 200 epochs. In the study, 83.72% accuracy was obtained on the training set and 72.38% accuracy on the validation data set. Recall, precision and prediction result for processed data is given in the Table 4.4.

Table 4.4 Top 4 emotion results for each image label

| Emotion | Angry | Hapy | Neutral | Sad |
|---|---|---|---|---|
| False Negative Rate | 5276 | 1253 | 4090 | 2012 |
| False Positive Rate | 0743 | 1328 | 1115 | 0620 |
| True Negative Rate | 9256 | 8671 | 8884 | 9379 |
| True Positive Rate | 4723 | 8746 | 5909 | 7987 |
| Precision | 6252 | 8329 | 5565 | 7243 |
| Recall | 4723 | 8746 | 5909 | 7987 |
| Accuracy | 8314 | 8704 | 8314 | 9143 |

The loss and accuracy graphs for each training process are given in Figure 4.8. Comparission with the Kaggle competition and state-of-the art values is given in Table 4.5.




a) b) c)

Figure 4.8 Loss and accuracy values for each dataset a) Original FER2013 dataset b) Processed FER2013 dataset c) 4 Selected emotion dataset

Table 4.5 FER-2013 emotion recognition accuracy comparison

| Study | Accuracy |
|---|---|
| (Mollahosseini, Chan, & Mahoor, 2016) | 66.4% |
| (Zhang et al., 2015) | 75.1% |
| (Tang, 2013) | 71.2% |
| (Ionescu, Popescu, & Grozea, 2013) | 67.48% |
| Our study – without pre-processing | 61.58% |
| Our study – on pre-processed and illustrated images | 56.21% |
| Our study – Top 4 emotion | 72.38% |

In the study, the highest accuracy rate for all three data sets was achieved for happy. In the study conducted on the processed data, a success rate of 85% was achieved and a success rate of 87% was obtained in the data set limited to 4 mental states. For all data sets, the second successful accuracy rate was obtained for sad. The accuracy of 70% on the processed data set and the accuracy of 75% on the data set limited to 4 emotion yielded an accuracy of 76% on the raw data set. 25.04% of the data set in which the study was carried out consisted of happy images and 11.51% consisted of sad images. By increasing the number of sample data describing each mental state, the accuracy rate can be increased to higher levels. Confusion matrices for unprocessed FER2013 data results is given in Figure 4.9, for processed FER2013 data results is given in Figure 4.10 and for top 4 emotion given in Figure 4.11.

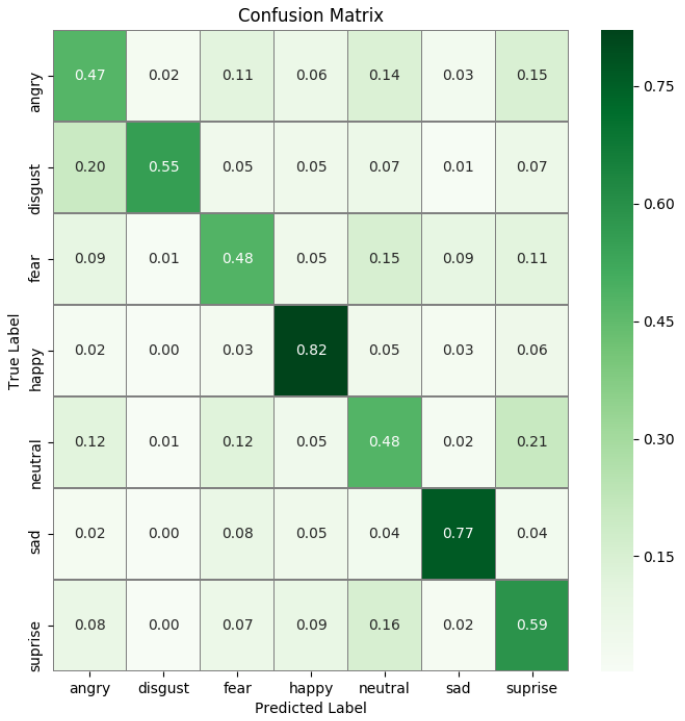



Figure 4.9 Confussion matrix for Original FER2013 dataset

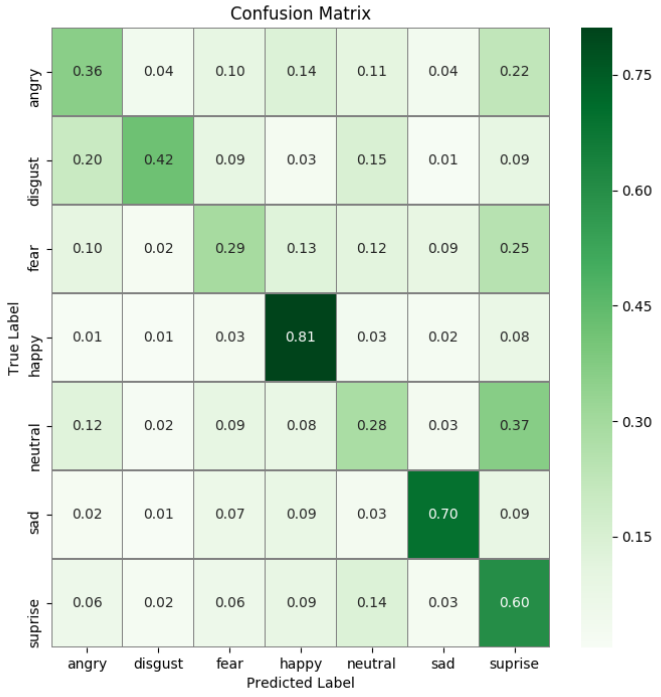



Figure 4.10 Confussion matrix for processed FER2013 dataset

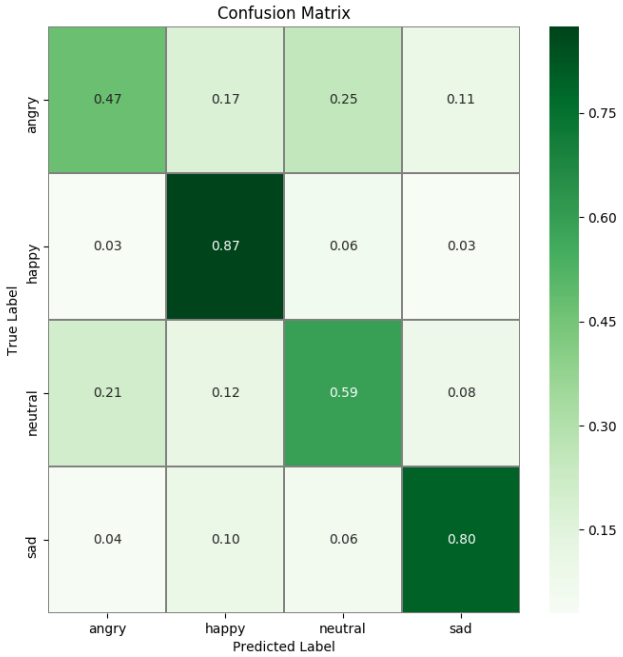



Figure 4.11 Confussion matrix for top 4 emotions of FER2013 dataset

# CHAPTER FIVE
# CONCLUSIONS AND FUTURE WORKS

As industry and business have developed, methods of increasing profitability have been sought other than reducing costs. Since value generation is dependent on personnel, especially in labor and service intensive sectors, productivity is directly related to profitability. Under these circumstances, many companies use different methods to increase the productivity of their employees. Employee monitoring is very important. However, alternative methods should be introduced to conventional methods.

With the fact that artificial intelligence is more and more involved in our lives with each passing day, it is possible to follow people's faces and analyze the mental state of them. For many companies, these methods can be used to monitor the mental state of their employees. However, due to the confidentiality of personal data, the collection and processing of these images needs to be designed with special consideration.

This thesis aims to design a emotion analysis system that works on closed circuit camera systems by performing emotion analysis on human face images which are free from personal data. With this system, it is aimed to establish a platform where companies can monitor their employees emotions individually and as a group in time manner.

The main objective of the application is to establish the classification according to the face emotions by processing the visual information that it reads from closed circuit camera systems of companies with a system consisting of different parts. The two main goals in this study are to provide a privacy-free data based system and to achieve an accuracy of more than 75% in emotion analysis.

For the purposes mentioned above, the application is based on three main structures. In the first stage, a system was developed which collects data from camera systems and makes face detection. This system performs face detection by reading data from

closed circuit camera systems and extracts facial landmark values from face. This results in a data free of personal data.

The second and third stages of the system perform preprocessing and classification processes. At this stage, instead of conventional artificial intelligence and machine learning systems, it is aimed to classify over the convolutional neural network. The main reason for this choice is the high accuracy obtained recently with the convolutional neural network. Because of these accuracy retaes which recently obtained in many study, convolutional neural network approach is preferred as the classification method.

Due to the preferred approach in the classification method, special preprocessing operations were required on the data which are collected from the closed circuit camera systems in the first stage. For this reason, face illustrations were created with the collected data. Prior to this procedure, face allignment was performed in order to increase the success rate.

In the last stage, model trainings were carried out with convolutional neural network over these face illustrations. During the model training, FER-2013 data set was used and training procedures were performed with different model structures. With the different convolutional neural network models, 56.21% accuracy was achieved on the whole data set and 72.38% for the basic 4 emotions. The accuracy rate which was obtained for the basic 4 emotions was close to that obtained accuracy rates in previous studies on the whole data set.

This study reveals that emotion recognition can be achieved by using the convolutional neural network approach on face illustrations based on facial landmark values. In this study, only certain parts of the face were converted to face illustration with specific drawing features. This method shows that mental state analysis can be performed on visuals that do not contain personal data. Different accuracy values were obtained with different illustration methods. Higher accuracy values can be obtained

in studies to be performed by showing different approaches in face illustration creation process.

In addition, changes in data set samples were reflected in accuracy rates. When the distribution of the pictures in the data set according to the emotions was examined, the highest success rates was obtained in the classes containing the most data. These results show that higher accuracy values can be achieved with model trainings with more sample-containing data.

All these results show that the proposed method is open to improvement and it is possible to achieve higher success rates with different studies.

Therefore, it is planned to obtain higher accuracy rates with different approaches in the future works of this study. The first future work planned for this study is to expand the data set to carry out trainings on singular and combinations of different academically accepted data sets. Therefore, it is planned to work on data sets containing a higher number of samples for both total and each emotion.

The next future work planned for this study is to examine the success rates with different variations of face illustrations. For this purpose, it is aimed to develop more meaningful data for convolutional neural network approach by improving the face illustration model used in this study. For this reason, it is aimed to add new face parts to the existing parts which are used in this study and try different methods in illustration processes of the face parts.

## REFERENCES


Abraham, R. (1999). The impact of emotional dissonance on organizational commitment and intention to turnover. *Journal of Psychology: Interdisciplinary and Applied*, *133*(4), 441–455.

Abramson, Y., Steux, B., & Ghorayeb, H. (2007). Yet Even Faster (YEF) real-time object detection. *International Journal of Intelligent Systems Technologies and Applications*, *2*(2/3), 102–112.

Ahonen, T., Hadid, A., & Pietikäinen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(28), 2037–2041.

Bhele, S. G., & Mankar, V. H. (2012). A Review Paper on Face Recognition Techniques. *International Journal of Advanced Research in Computer Engineering & Technology*, *1*(8), 2278–1323.

Bouhabba, E. M., Shafie, A. A., & Akmeliawati, R. (2011). Support vector machine for face emotion detection on real time basis. *2011 4th International Conference on Mechatronics: Integrated Engineering for Industrial and Societal Development, ICOM'11 - Conference Proceedings*, 1–6.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, *1*, 886–893.

Déniz, O., Bueno, G., Salido, J., & De La Torre, F. (2011). Face recognition using Histograms of Oriented Gradients. *Pattern Recognition Letters*, *32*(12), 1598–1603.

Ekman, P. (1970). Universal-Facial-Expressions-of-Emotion. *California Mental Health*, *8*(4), 151–158.

Ekman, P. (1992). An Argument for Basic Emotions. *Cognition and Emotion*, *6*(3–4),

169–200.

Ekman, P., & Friesen, W. V. (1969). The repertoire of nonverbal behavior: Categories, origins, usage, and coding. *Semiotica*, *1*(1), 49–98.

Ekman, P., & Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, *17*(2), 124–129.

El Naqa, I., & Murphy, M. J. (2015). What Is Machine Learning? In *Machine Learning in Radiation Oncology* (3–11). Chan : Springer International Publishing.

Farfade, S. S., Saberian, M., & Li, L. J. (2015). Multi-view face detection using Deep convolutional neural networks. *ICMR 2015 - Proceedings of the 2015 ACM International Conference on Multimedia Retrieval*, 643–650.

Fisher, C. D., & Noble, C. S. (2004). A within-person examination of correlates of performance and emotions while working. *Human Performance*, *17*(2), 145–168.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, *36*(4), 193–202.

Goodfellow, I. J., Erhan, D., Luc Carrier, P., Courville, A., Mirza, M., & Hamner, B. (2015). Challenges in representation learning: A report on three machine learning contests. *Neural Networks*, *64*(C), 59–63.

Happy, S. L., George, A., & Routray, A. (2012). A real time facial expression classification system using local binary patterns. In *2012 4th International conference on intelligent human computer interaction (IHCI)* (1–5).

Hinkin, T. R., & Tracey, J. B. (2000). The Cost of Turnover. *Cornell Hotel and Restaurant Administration Quarterly*, *41*(3), 14–21.

Hsu, R. L., Abdel-Mottaleb, M., & Jain, A. K. (2002). Face detection in color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(5), 696–706.

Ionescu, R. T., Popescu, M., & Grozea, C. (2013). Local Learning to Improve Bag of Visual Words Model for Facial Expression Recognition. *Workshop on Challenges in Representation Learning, ICML*, 1–6.

Jain, S., Durgesh, M., & Ramesh, T. (2016). Facial expression recognition using variants of LBP and classifier fusion. In *Advances in Intelligent Systems and Computing*, *408*, 725–732.

Lawrence, S., Giles, C. L., Tsoi, A. C., & Back, A. D. (1997). Face recognition: A convolutional neural-network approach. *IEEE Transactions on Neural Networks*, *8*(1), 98–113.

Lesiuk, T. (2005). The effect of music listening on work performance. *Psychology of Music*, *33*(2), 173–191.

Li, H., Lin, Z., Shen, X., & Brandt, J. (2015). A convolutional neural network cascade for face detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5325–5334.

Lienhart, R., & Maydt, J. (2002). An extended set of Haar-like features for rapid object detection. In *Proceedings. international conference on image processing*, 900–903.

Mcculloch, W. S., Lettvin, J. Y., Pitts, W. H., & Dell, P. C. (1952). An electrical hypothesis of central inhibition and facilitation. *Research Publications - Association for Research in Nervous and Mental Disease*, *30*, 87–97.

Miner, A. G., & Glomb, T. M. (2010). State mood, task performance, and behavior at work: A within-persons approach. *Organizational Behavior and Human Decision Processes*, *112*(1), 43–57.

Mollahosseini, A., Chan, D., & Mahoor, M. H. (2016). Going deeper in facial expression recognition using deep neural networks. *2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016*, 1–10.

Nazir, M., Jan, Z., & Sajjad, M. (2018). Facial expression recognition using histogram

of oriented gradients based transformed features. *Cluster Computing*, *21*(1), 539–548.

Nielsen, M. (2015). Chapter 3 - Improving the way neural networks learn. In *Neural Networks and Deep Learning* (1–130). San Francisco, CA, USA: Determination press.

Ojala, T., Pietikäinen, M., & Harwood, D. (1994). Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. *Proceedings - International Conference on Pattern Recognition*, *3*, 582–585.

Owusu, E., Zhan, Y., & Mao, Q. R. (2014). A neural-AdaBoost based facial expression recognition system. *Expert Systems with Applications*, *41*(7), 3383–3390.

Plutchik, R. (1980). A General Psychoevolutionary Theory of Emotion. In *Theories of Emotion* (3–33). Amsterdam : Academic press.

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, *1*(1), 81–106.

Ramos, C., Augusto, J. C., & Shapiro, D. (2008). Ambient intelligencethe next step for artificial intelligence. *IEEE Intelligent Systems*, *23*(2), 15–18.

Rowley, H. A., Baluja, S., & Kanade, T. (1998a). Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(1), 23–38.

Rowley, H. A., Baluja, S., & Kanade, T. (1998b). Rotation invariant neural network-based face detection. In *Proceedings. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 38–44.

Samuel, A. L. (2000). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, *44*(1–2), 207–219.

Sebe, N., Lew, M. S., Cohen, I., Sun, Y., Gevers, T., & Huang, T. S. (2004). Authentic facial expression analysis. *Proceedings - Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, 517–522.

Sun, Y., Wang, X., & Tang, X. (2015). Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2892–2900.

Tang, Y. (2013). *Deep Learning using Linear Support Vector Machines*. Retrieved September 14, 2019, from https://arxiv.org/pdf/1306.0239v4.pdf

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 511–518.

Werghi, N., Tortorici, C., Berretti, S., & Del Bimbo, A. (2016). Boosting 3D LBP-Based Face Recognition by Fusing Shape and Texture Descriptors on the Mesh. *IEEE Transactions on Information Forensics and Security*, *11*(5), 964–979.

Wright, T. A., & Bonett, D. G. (2007). Job satisfaction and psychological well-being as nonadditive predictors of workplace turnove. *Journal of Management*, *33*(2), 141–160.

Zelenski, J. M., Murphy, S. A., & Jenkins, D. A. (2008). The happy-productive worker thesis revisited. *Journal of Happiness Studies*, *9*(4), 521–537.

Zhang, Z., Luo, P., Loy, C. C., & Tang, X. (2015). Learning social relation traits from face images. *Proceedings of the IEEE International Conference on Computer Vision, 2015 Inter*, 3631–3639.