TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Ismayil Aliyev 184625IASM

FACIAL EMOTIONAL ANALYSIS BASED ON A SYSTEM ON CHIP

Master's Thesis

Supervisor
Kristina Vasiljeva
Associate Professor

Tallinn 2020

Author's declaration of originality

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

Author: Ismayil Aliyev ....

(signature)

Date: Month Day, Year

Abstract

Nowadays, human machine interaction field is growing tremendously and emotions could be understood based on text, fluctuations of voice waves and facial expressions. Analyzing Human Emotions has multiple applications in various fields namely medical purpose, surveillance, security, entertainment and ending with marketing research. Application of current work is marketing analysis for digital banner ads to get instant feedback from potential customers who glance on LED screens. Modern-day technology is advanced enough to help in analyzing the Facial Emotions of people based on real-time video recording by using computer vision algorithms and accuracy is measured in confusion matrix. However, even with technological advancements, it is not easy to perform necessary computations by using System on Chip. In the current thesis work, the author built systems for Facial Sentiment Analysis of people with different Systems on Chips, which included Raspberry Pi Zero, Raspberry Pi 3 and Raspberry Pi 4. All Systems on Chips are powered by Linux OS, specifically Raspbian. Python 3.8 is a working programming language. OpenCV, NumPy, PIL, Requests, Scikit-Learn, YOLO and TensorFlow are used as python libraries.

The thesis is written in English and it contains 48 pages of text, 6 chapters, 48 figures.

Source code could be found in the following link.

Thesis project repository could be found in following link https://github.com/Ismayil94/ThesisProject

List of abbreviations and terms

AI	Artificial Intelligence
API	Application Programming Interface
AWS	Amazon Web Services
CPU	Central Processing Unit
CNN	Convolution Neural Network
IDE	Integrated Development Environment
IOT	Internet Of Things
ML	Machine Learning
MQTT	MQ Telemetry Transport
YOLO	You Only Look Once
RGB	Red Green Blue Channels
SGD	Stochastic Gradient Descent
GPU	Graphics Processing Unit
PIP	Python Package Index
CPU	Central Process Unit
GPU	Graphics processing unit
GUI	Graphical User Interface
IDE	Integrated Development Environment
IP	Internet Protocol
LED	Light Emitting Diode
OS	Operating System
RAM	Random Access Memory
SoC	System On Chip
SSH	Secure Shell
UDP	User Datagram Protocol
VNC	Virtual Network Computing
Wi-Fi	Wireless Local Area Networking
VENV	Virtual environment
PIP	Python Package Index

Table of Contents

List of Figures vi

List of Tables viii

1 Introduction **1**

1.1 Motivation 1

1.2 Problem 1

1.3 Solution 2

1.3.1 Research 2

1.4 Market 3

1.5 Emotion Models 4

1.6 Task Objectives and Definitions 6

1.6.1 Pipeline 7

1.6.2 Facial Emotions Databases 8

1.7 System on Chip Description 9

1.8 Thesis Objectives 11

1.9 Thesis Structure 11

2 Background **13**

2.1 Artificial Intelligence and Deep Learning 13

2.1.1 Machine Learning 15

2.1.2 Neural Networks 17

2.2 Fundamentals of Image Processing 19

2.2.1 Colored Images 20

2.2.2 Operations on Images with Python Image Library 21

2.3 Software Dependencies 23

2.3.1 Python 24

2.3.2 Virtual Environment 24

2.3.3 Numpy 25

2.3.4 Pandas 25

2.3.5 Tensorflow 25

2.3.6 Tensorflow Lite 25

2.4 Conclusion 26

3 Methodology and Design **27**

3.1 Pipeline of Facial Emotional Analysis 27

3.1.1 Face Detection 28
3.1.2 Active Shape Model 30
3.1.3 Facial Emotional Recognition with Convolutional Neural Networks and AdaBoost 31
3.2 Raspberry Pi Development 32
3.2.1 Training Model 32
3.2.2 Raspberry Pi Configurations 36
3.3 Summary 42

4 Evaluation **43**
4.1 Confusion Matrix 43
4.1.1 TensorFlow 44
4.1.2 TensorFlow Lite 46
4.2 Inference Time 47
4.3 Conclusion 48

5 Future Work **49**
5.0.1 System Description 49
5.0.2 Django for Web Application Development 50
5.0.3 Heatmap 52

6 Summary **53**
6.0.1 Limitations 53
6.0.2 Conclusion 53

Bibliography **54**

List of Figures

1 *Digital Signage Market Outlook - 2026 [2]* 3

2 *Emotion Detection and Recognition market [3]* 4

3 *Plutchik's Wheel of Emotions [6]* 5

4 *Ekman's model of basic emotion division [7]* 6

5 *Procedure used in conventional Facial Emotional Recognition [8]* 7

6 *Procedure of CNN-based Facial Emotional Recognition [8]* 7

7 *Databases for facial emotional recognition [8]* 9

8 *Raspberry Pi 3 and Pi Zero Specs Comparison [9]* 10

9 *Raspberry Pi 3 and Pi 4 Specs Comparison [10]* 11

10 *Traditional Approach of programming [14]* 14

11 *Machine Learning approach [14]* 15

12 *Machine Learning Tree and branches [15]* 15

13 *Workflow of Machine Learning process [15]* 16

14 *Examples of Supervised and Unsupervised Learning [14]* 17

15 *Biological Neuron and Artificial Neuron [13]* 17

16 *Perceptron structure [16]* 18

17 *Neural Networks Structure [17]* 19

18 *Image Data [18]* 20

19 *Image Data [18]* 20

20 *Decomposition method of RGB images. (1) Original Lena image; (2) Red component; (3) Green component; (4) Blue component [19]* 21

21 *Python Imaging Library operations [21]* 22

22 *Python Imaging Library operations [21]* 22

23 *Python Imaging Library operations [21]* 23

24 *Python Imaging Library operations [21]* 23

25 *Tensorflow Lite Architecture [23]* 26

26 *Pipeline of facial emotional recogntion [17]* 27

27 *Haar-Like Features [25]* 28

28 *Haar-Like Features [26]* 29

29 *: Good landmarks are points of high curvature or junctions [27]* 30

30 *Active Shape Model Search progressing [27]* 31

31 *Description of Facial Muscles involved in expressing emotions [28]* . . . 32

32 *Accessing Dataset* 33

33 *Training Model with Deep Learning* 33

34 *CNN structure* 34
35 *Training Model with Deep Learning* 34
36 *Evaluation the Model* 35
37 *Evaluation the Model* 35
38 *Confusion Matrix* 35
39 *Confusion Matrix* 36
40 *Downloading Operating System's Image [29]* 37
41 *Burn OS image to micro SD card [30]* 37
42 *Connect micro SD card to Raspberry Pi [31]* 38
43 *Configuring WiFi [31]* 38
44 *Use VNC viewer for remote control [32]* 39
45 *Pi Camera Module [33]* 40
46 *Enabling Pi Camera [33]* 40

47 *Confusion matrix. Exemplified CM with the formulas of precision (PR), recall (RE), accuracy (CA), and F1-measure [37]* 43
48 *Confusion Matrix for Raspberry Pi 3 on TensorFlow* 44
49 *Performance on Raspberry Pi 3 on TensorFlow* 45
50 *Confusion Matrix for Raspberry PI4 on TensorFlow* 45
51 *Performance on Raspberry PI4 on TensorFlow* 45
52 *Performance on Raspberry Pi 3 on TensorFlow Lite* 46
53 *Performance on Raspberry Pi 3 on TensorFlow Lite* 46
54 *Performance on Raspberry Pi 4 on TensorFlow Lite* 47
55 *Performance on Raspberry Pi 4 on TensorFlow Lite* 47
56 *Inference Time Comparison [39]* 48

57 *The workflow of the Facial Emotions Recognition System.* 49
58 *Django Web Development Framework Model View Template approach [40]* 51
59 *Example of Heatmap which indicates the attention of people [42]* 52

List of Tables

1. Introduction

1.1 Motivation

Facial emotional analysis is very popular trend and done for multiple reasons, because it has scientific value as well as business value. Despite having broaden implications of facial emotional analysis it is main goal is to make instant feedback with people and make analysis on how do they like some product, solution, medical treatment etc.

In this thesis, the application of facial emotional analysis is making A/B analysis of various designs of billboard ads.

End-goal is to build a device which will make public surveillance and make facial emotional analysis of people who glance at it. Attention Heatmap will be generated based on gathered data.

Considering that, the smart device is built by using microprocessor with relatively not strong computation power, performing facial emotional analysis is not an easy task, various training models and dataset databases will be tested in thesis.

In spite of the fact that there are tons of research made in real-time facial emotional analysis, there are no research which compares performance of various technological approaches by using different systems on chip designs.

1.2 Problem

Some ads might irritate people and scare off potential customers by perceiving bad branding. While using outdoor and indoor advertisements, businesses are not aware of the information on which design suits best and how many people glanced on the ads of their products.

The main features of the products sometimes are designed or put wrong in promo materials. They do not know the gender, age and emotions of customers they had looking at different parts of banner ads, suitable time spot for the advertisement and so on.

It is a long process to negotiate with the advertisement companies in order to run ads on different indoor/outdoor screens.

1.3 Solution

The Solution suggested for mentioned issue consists of a web platform and smart device. The device is a microprocessor with the external camera on it, that runs AI software.

The third component of solutions is cloud computing. The web platform will have a user profile, with the help of that users will log in, run ads campaigns and get heatmap of attendance indicating what emotions people had while looking at banner ads.

The device will control LED screens by uploading an image/video file to it, and will run AI program to perform real-time facial emotions analysis. The thesis describes research done to find best and least costly microprocessor to implement the solution.

1.3.1 Research

Research questions and research method should be defined in order to answer to any research.

Research Questions

This thesis will describe the research which answers following research questions:

- Are embedded systems specifically raspberry microprocessors good enough to perform facial emotional analysis ?
- What microprocessor is enough to perform facial emotional analysis among raspberry pi zero, 3 and 4?

Research Method

Research method should be defined in order to answer to any research question. A research method is a strategy used in order to answer to research questions. In our case, the answers will be found by doing **Experiments** [1].

1.4 Market

The global digital signage market size is expected to grow with 8.1% annually and reach $32.12 billions in 2026 [2]. A digital signage is an electronic display that running advertisements with various content. LED/LCD displays are used as the hardware component which are located in restaurants, shopping malls, business centers, fitness centers etc.

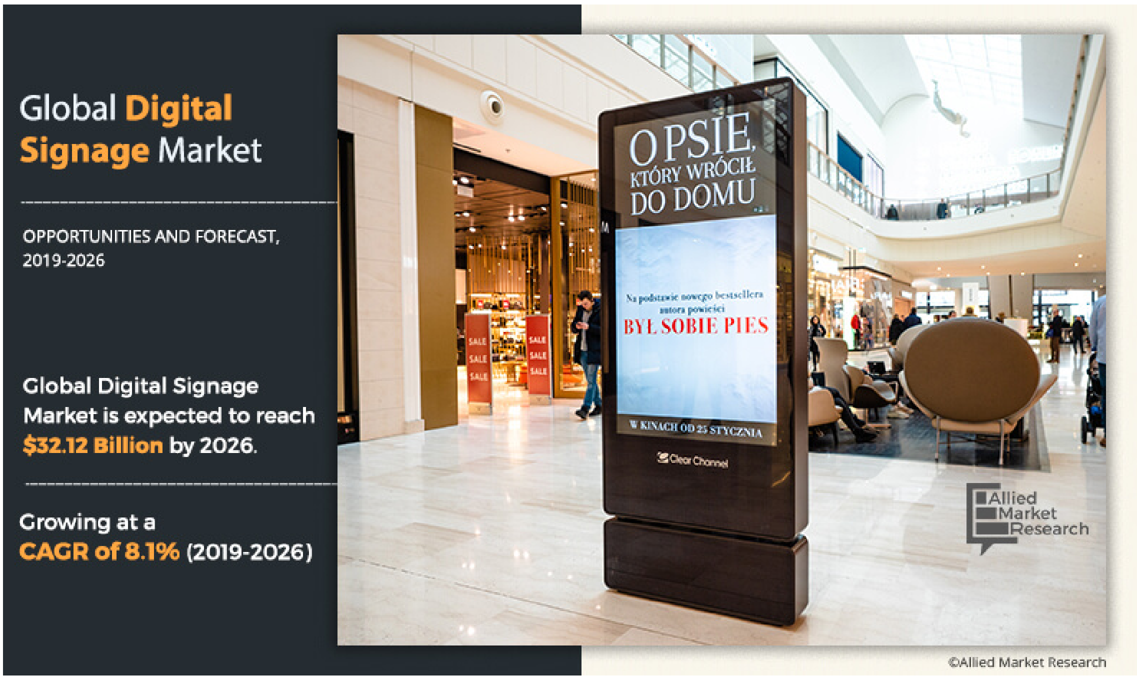

Global Digital Signage Market
OPPORTUNITIES AND FORECAST, 2019-2026
Global Digital Signage Market is expected to reach $32.12 Billion by 2026.
Growing at a CAGR of 8.1% (2019-2026)
O PSIE, KTÓRY WRÓCIŁ DO DOMU
Na podstawie nowego bestsellera autora powieści
BYŁ SOBIE PIES
W KINACH OD 25 STYCZNIA
Clear Channel
SALE
Allied Market Research
©Allied Market Research

Figure 1. *Digital Signage Market Outlook - 2026 [2]*

In addition to the growth of digital signage market, the global emotion detection and recognition market size is expected to double by 2024 with compound annual growth rate of 21.0% [3].

Factors such as the increasing of need for real time feedback, biometric systems for authentications are driving the market up.

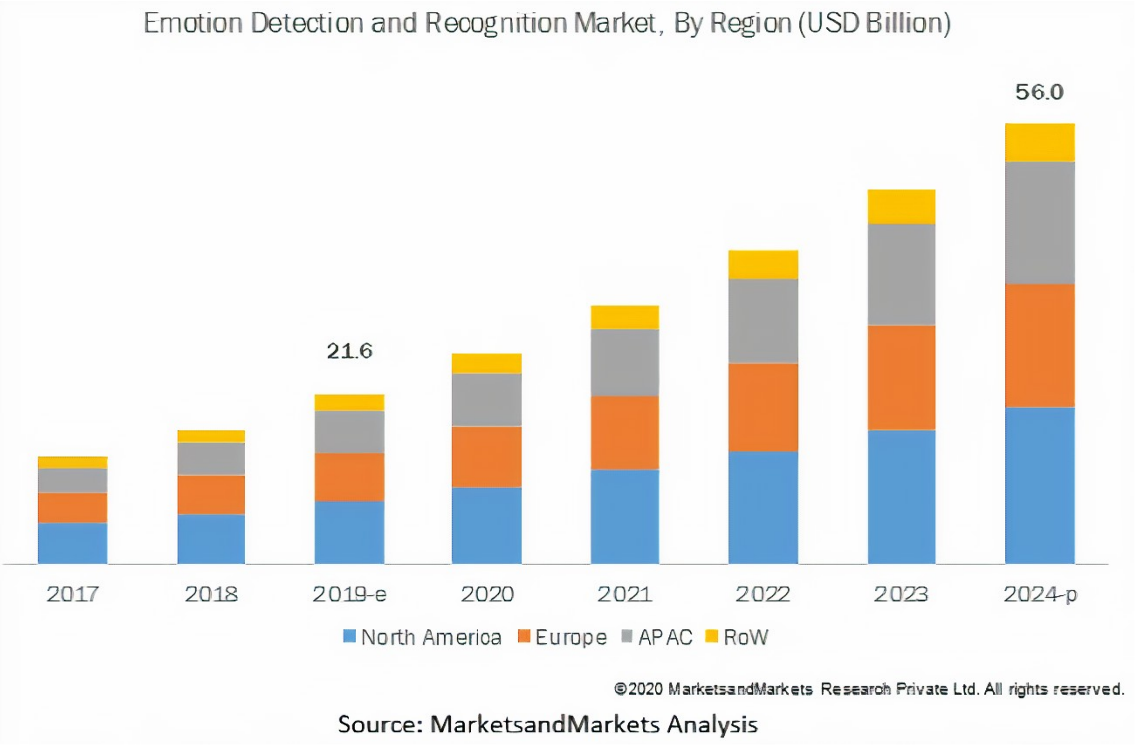


Figure 2. *Emotion Detection and Recognition market [3]*

1.5 Emotion Models

Emotions are complex and there is separate science branch dedicated to researching emotions and emotional experience, and classifying emotions which is called as Affective science.

In general, there are categorical and dimensional emotional model categories. The categorical model uses six basic emotion classes namely anger, disgust, fear, joy, sadness, and surprise. Every emotion expressed after by some features after some circumstances or reactions. In contrast, Dimensional models of emotion suggest that one common interconnected neurophysiological system is responsible for emotions [4], [5].

There were various tries to classify emotions but the most effective work was done by famous psychologist Robert Plutchik. Based on theory of Psychologist Robert Plutchik, there are 8 basic emotions namely joy, trust, fear, surprise, sadness, anticipation, anger and disgust. There is model called Plutchik's wheel of emotions illustrating 8 basic emotions and emotions which are derivative from those basic emotions.

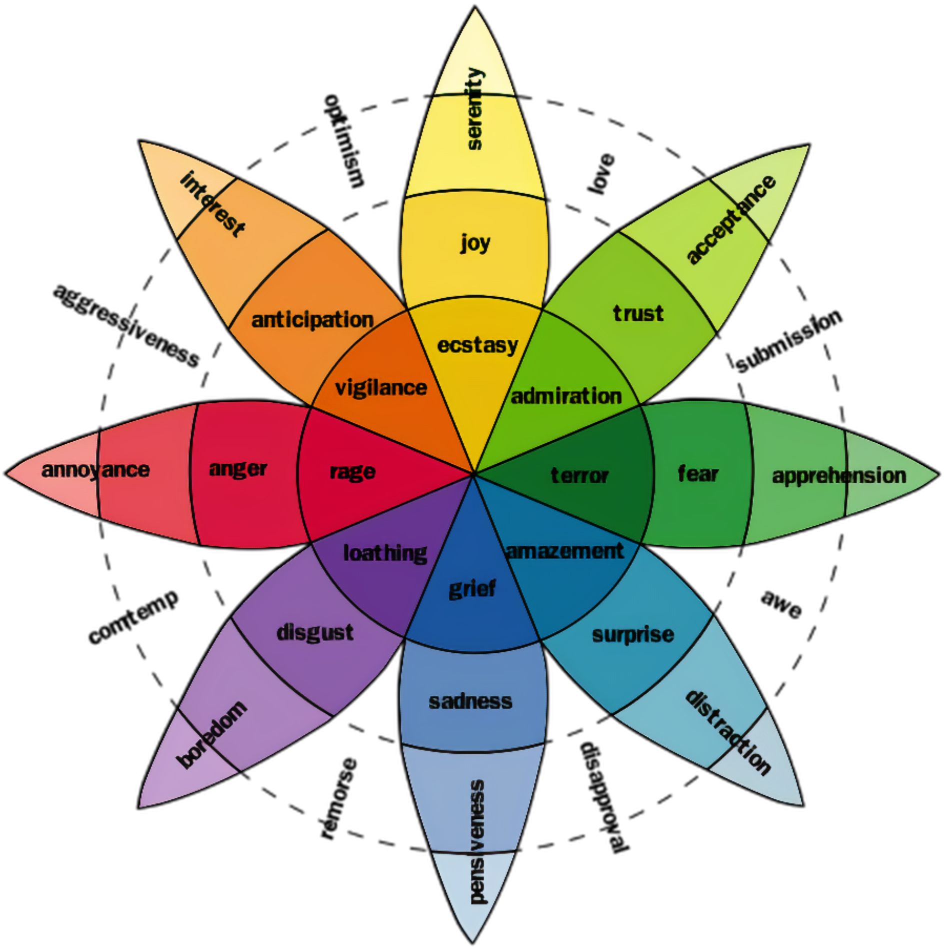


Figure 3. *Plutchik's Wheel of Emotions [6]*

Oppositely, there is theory developed by American psychologists Paul Ekman and Wallace V. Friesen claiming that there are six basic emotions.
Those six basic emotions are developed after observing and researching emotions of peoples in Fori tribe in Papua New Guinea in 1972 [7].

Those emotions are following: Anger, Disgust, Fear, Happiness, Sadness, Surprise.

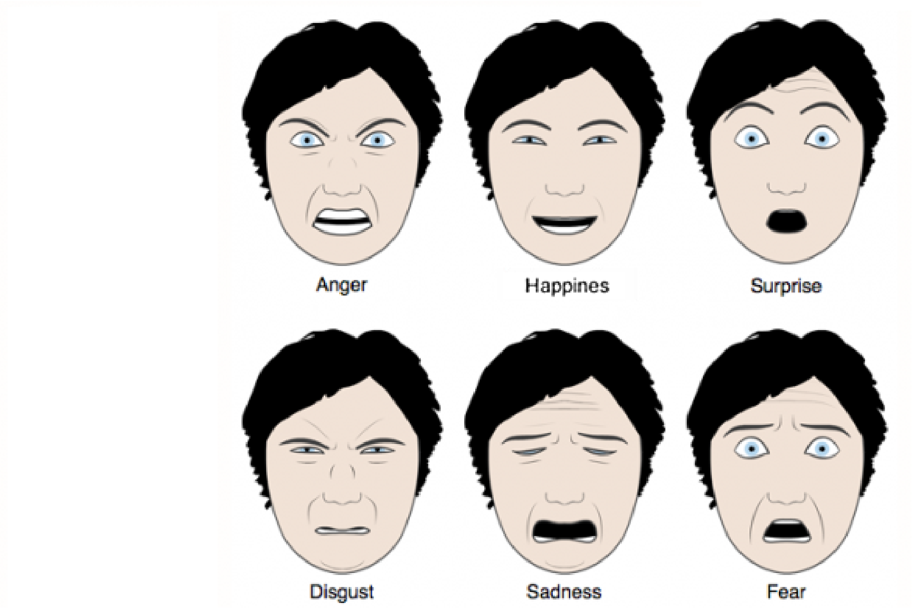


Figure 4. *Ekman's model of basic emotion division [7]*

Also interpretation of emotions based on features might be different depending on context. For example, in the education context, five categories namely boredom, confusion, joy, flow, and frustration proposed to describe affect of students. Because students are not feeling basic feelings of fear or disgust but instead they may experience boredom or delight.

1.6 Task Objectives and Definitions

The work is done with Raspberry Pi Zero, Raspberry Pi 3 and Raspberry Pi 4 by using Raspberry Pi Camera and web streaming USB 2.0 camera, to analyze the facial emotions of people who glance at the device connected to LED/LCD screens.

Artificial Intelligence software is built based on three main algorithms implemented in python using the OpenCV library.

Eigen Face Algorithm, Fisher Face Algorithm, Local Binary Patterns Histogram and by using Convolutional Neural Networks. All combinations will be used and tested to find the perfect performance. Evaluation of classification model on Rapberry Pi Zero, 3 and 4 is done and described in thesis.

The purpose of whole system is to gather and analyse emotions of people who glance on digital banner ads.

1.6.1 Pipeline

In this subsection, the pipelines of facial emotional recognition are described.

There are three steps in conventional facial emotional recognition methods:

- Getting Images as an Input. In our case it is real-time, from video stream.
- Face detection. Face detection happens by finding specific features like finding eyes and nose.
- Feature extraction. Various features are extracted and separated from the faces.
- Classification of emotions. Finally, based on extracted features the trained model makes classification of emotions.

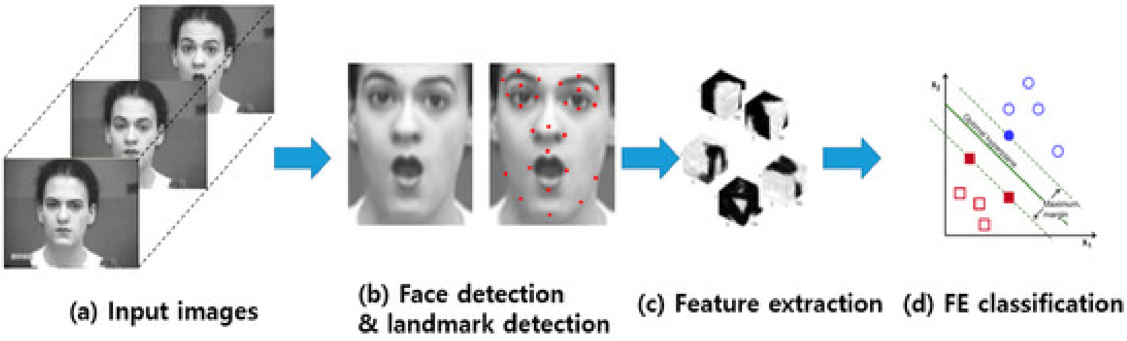


Figure 5. *Procedure used in conventional Facial Emotional Recognition [8]*

Oppositely to traditional methods, the pipeline of using Convolutional Neural Networks consist of following: the input image is went trough checks of convolution layers to generate a feature map. Leading to recognizing and classifying the facial expressions.

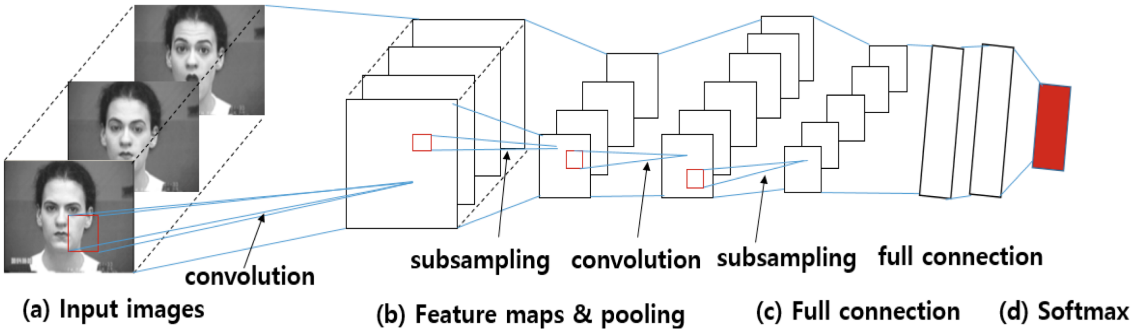


Figure 6. *Procedure of CNN-based Facial Emotional Recognition [8]*

1.6.2 Facial Emotions Databases

In order to train the Facial Emotional Recognition model various numerous databases might be used. Both 2D and 3D images are used in most of them.

The Extended Cohn-Kanade Dataset has 593 videos on emotions. The age range is from 18 to 30 years, with majority of females. Those images have pixel resolutions of 640 * 480 and 640 * 490.

Compound Emotion (CE) contains 5060 images with 22 categories of both basic and compound emotions. Most races are included, with namely Caucasian, Asian, African, and Hispanic. The photos are color images with a pixel resolution of 3000 * 4000.

Denver Intensity of Spontaneous Facial Action Database (DISFA) consists of 130,000 video frames at high resolution of 1024 * 768 and 27 adult subjects with different ethnicities.

Binghamton-Pittsburgh 3D Dynamic Spontaneous is a 3D video database that includes a group of 41 young adults of 23 women and 18 men of 18–29 years age range. 11 Asian, 6 African-American, 4 Hispanic, and more than 20 Euro-Americans. The size of facial images is 1040 pixels * 1329 pixels.

MMI database consists of over 2900 video sequences and high-resolution images. It contains a total of 238 videos with 28 subjects, both males and females. The size of facial images is 720 pixels * 576 pixels.

The Karolinska Directed Emotional Face database contains 4900 images of human facial emotional expressions containing of 70 individuals, each displaying seven different emotional expressions with the size of image is 562 pixels * 762 pixels.

Japanese Female Facial Expressions database contains of 213 images with seven facial emotions. The size of images is 256 * 256 pixels. Extended Yale B face database consists of a set of 16,128 facial images contains 28 distinct subjects for 576 viewing conditions. The size of each facial image is 320 * 243 pixels.

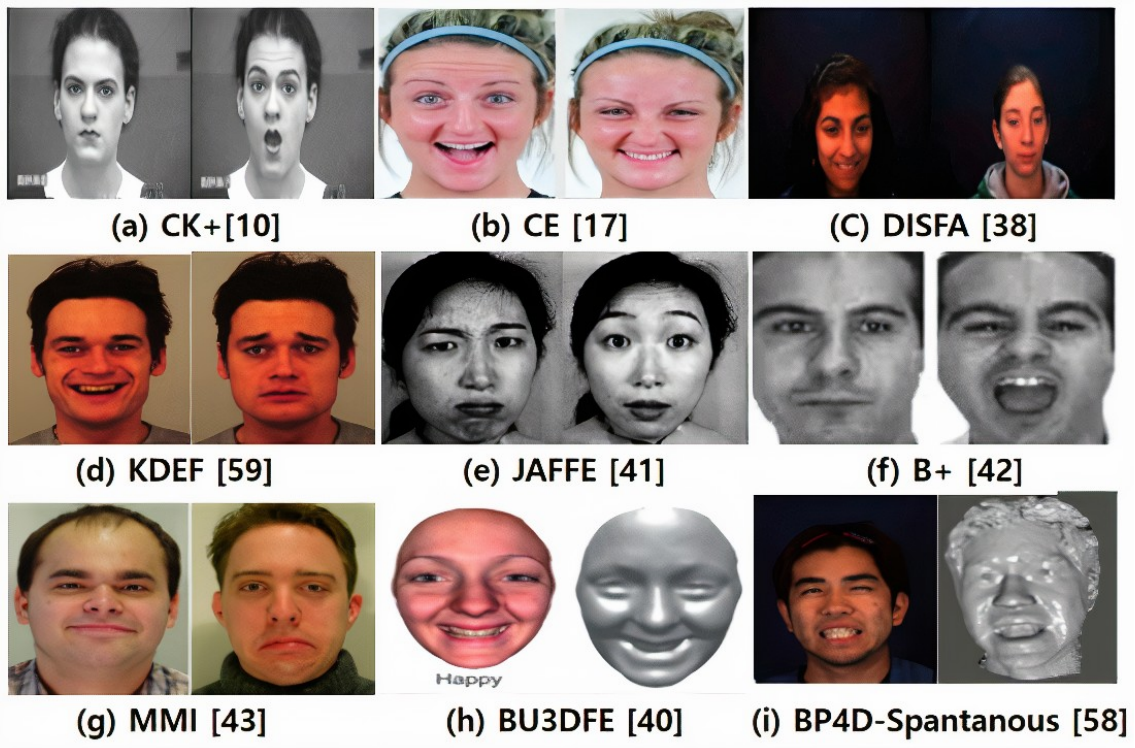


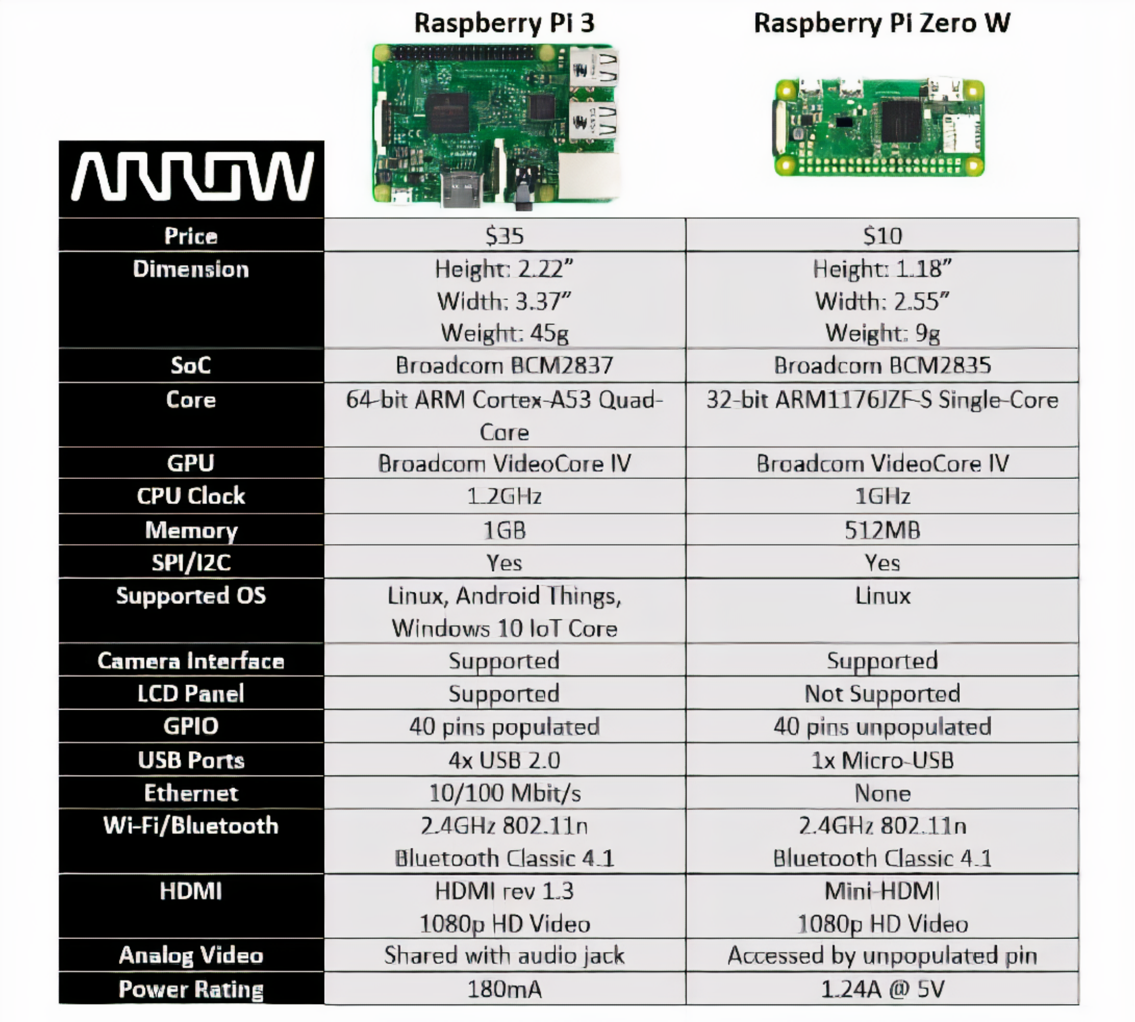
Figure 7. *Databases for facial emotional recognition [8]*

1.7 System on Chip Description

As it mentioned previously, Raspberry Pi 3, Raspberry Pi Zero and Raspberry Pi 4 microprocessors will be used and the precision and performance of real-time facial emotions analysis will be compared on those Microprocessors.

Raspberry Pi is a small single-board computer designed by the Raspberry Pi Foundation in the United Kingdom initially aimed to promote the teaching of computer science. The majority of Pis are made in a Sony factory in Wales, while others are made in China and Japan

The association behind Raspberry Pi consist of two branches. The initial two models are created by the Raspberry Pi Foundation.



	Raspberry Pi 3	Raspberry Pi Zero W
Price	$35	$10
Dimension	Height: 2.22” Width: 3.37” Weight: 45g	Height: 1.18” Width: 2.55” Weight: 9g
SoC	Broadcom BCM2837	Broadcom BCM2835
Core	64-bit ARM Cortex-A53 Quad-Core	32-bit ARM1176JZF-S Single-Core
GPU	Broadcom VideoCore IV	Broadcom VideoCore IV
CPU Clock	1.2GHz	1GHz
Memory	1GB	512MB
SPI/I2C	Yes	Yes
Supported OS	Linux, Android Things, Windows 10 IoT Core	Linux
Camera Interface	Supported	Supported
LCD Panel	Supported	Not Supported
GPIO	40 pins populated	40 pins unpopulated
USB Ports	4x USB 2.0	1x Micro-USB
Ethernet	10/100 Mbit/s	None
Wi-Fi/Bluetooth	2.4GHz 802.11n Bluetooth Classic 4.1	2.4GHz 802.11n Bluetooth Classic 4.1
HDMI	HDMI rev 1.3 1080p HD Video	Mini-HDMI 1080p HD Video
Analog Video	Shared with audio jack	Accessed by unpopulated pin
Power Rating	180mA	1.24A @ 5V

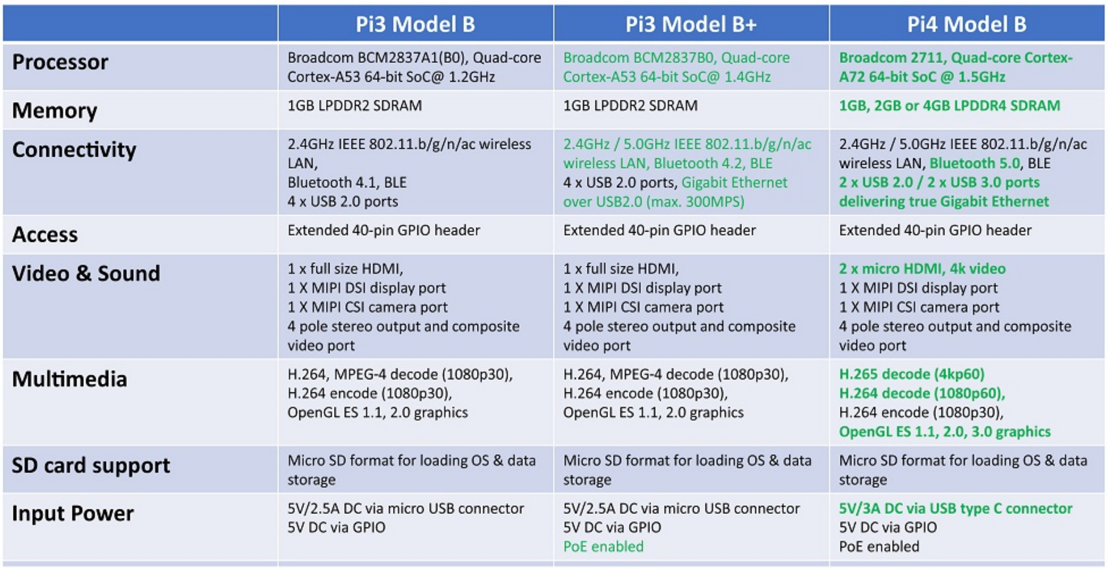
Figure 8. *Raspberry Pi 3 and Pi Zero Specs Comparison [9]*

The table above demonstrate the difference and similarities for Raspberry PI 3 and Raspberry PI Zero.

Clearly, Raspberry Pi 3 outnumber Raspberry Pi zero in performance and memory, but both can be used for our goal of performing real-time facial emotional analysis [9].

The latest product of Raspberry Pi computers is Raspberry Pi 4 Model B and it offers increases in processor speed, memory with remaining same amount of power consumption [10].

You can find detailed comparison of Raspberry Pi 4 and Pi 3 below.



	Pi3 Model B	Pi3 Model B+	Pi4 Model B
Processor	Broadcom BCM2837A1(B0), Quad-core Cortex-A53 64-bit SoC@ 1.2GHz	Broadcom BCM2837B0, Quad-core Cortex-A53 64-bit SoC@ 1.4GHz	**Broadcom 2711, Quad-core Cortex-A72 64-bit SoC @ 1.5GHz**
Memory	1GB LPDDR2 SDRAM	1GB LPDDR2 SDRAM	**1GB, 2GB or 4GB LPDDR4 SDRAM**
Connectivity	2.4GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.1, BLE 4 x USB 2.0 ports	2.4GHz / 5.0GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE 4 x USB 2.0 ports, Gigabit Ethernet over USB2.0 (max. 300MPS)	2.4GHz / 5.0GHz IEEE 802.11.b/g/n/ac wireless LAN, **Bluetooth 5.0**, BLE **2 x USB 2.0 / 2 x USB 3.0 ports delivering true Gigabit Ethernet**
Access	Extended 40-pin GPIO header	Extended 40-pin GPIO header	Extended 40-pin GPIO header
Video & Sound	1 x full size HDMI, 1 X MIPI DSI display port 1 X MIPI CSI camera port 4 pole stereo output and composite video port	1 x full size HDMI, 1 X MIPI DSI display port 1 X MIPI CSI camera port 4 pole stereo output and composite video port	**2 x micro HDMI, 4k video** 1 X MIPI DSI display port 1 X MIPI CSI camera port 4 pole stereo output and composite video port
Multimedia	H.264, MPEG-4 decode (1080p30), H.264 encode (1080p30), OpenGL ES 1.1, 2.0 graphics	H.264, MPEG-4 decode (1080p30), H.264 encode (1080p30), OpenGL ES 1.1, 2.0 graphics	**H.265 decode (4kp60)** **H.264 decode (1080p60),** H.264 encode (1080p30), **OpenGL ES 1.1, 2.0, 3.0 graphics**
SD card support	Micro SD format for loading OS & data storage	Micro SD format for loading OS & data storage	Micro SD format for loading OS & data storage
Input Power	5V/2.5A DC via micro USB connector 5V DC via GPIO	5V/2.5A DC via micro USB connector 5V DC via GPIO PoE enabled	**5V/3A DC via USB type C connector** 5V DC via GPIO PoE enabled

Figure 9. *Raspberry Pi 3 and Pi 4 Specs Comparison [10]*

It's claimed that Raspberry Pi 4 can be used as PC for daily tasks.

1.8 Thesis Objectives

Current thesis has two goals listed below:

- Describe the System for facial emotional analysis based on System on Chip. The system which consist of web application, smart device analysing emotions of people in real-time by running deep learning software.

- Make evaluation of performance of facial emotional analysis by using different microprocessors, Raspberry Zero, Pi 3 and Pi 4. Accuracy, Precision and Recall and build Confusion Matrix based on False Positive and False Negative cases and the performance of computation on all three of microprocessors is done.

1.9 Thesis Structure

Thesis consists of five sections.

It starts with the Introduction section which gives general view of work done and structure of built system. Motivation of research, market growth and opportunities are also described in Introduction section.

The analysis of state of the art based on various research papers will be mentioned and analysed in chapter two named as Background.

Chapter 3 will include the methodology of doing facial emotional analysis on system on chips and the design of the system. By using Raspberry Pi Zero, Pi 3 and Pi 4 by using Pi Camera and web streaming camera.

Artificial Intelligence software is built based on three main algorithms implemented in python using the OpenCV library. Eigen Face Algorithm, Fisher Face Algorithm, Local Binary Patterns Histogram and by using Convolutional Neural Networks.

Evaluation of performance of deep learning model on raspberry zero, pi 3 and pi 4 and their comparison will be illustrated and described in chapter 4 named as Evaluation and future work.

In addition, future improvements and applications of the project and research will be discussed in the same section. The final chapter, as name indicates will include the overall conclusion.

2. Background

Face emotional analysis is a key areas of pattern recognition, computer vision and artificial intelligence. This chapter will illuminate current state of art related to Deep Learning, Computer Vision and will describe libraries and frameworks used in those fields.

2.1 Artificial Intelligence and Deep Learning

Artificial Intelligence is the broaden term of developing intelligent machines that mimics humans and human behavior, basically this domain aimed to make machines smart. Whereas Machine Learning is a subset of AI, meaning of the systems which can learn by themselves by using data.

Deep Learning is a subset of Machine Learning and it's using larger data sets and working with neural networks [11], [12], [13].

The traditional programming is done by analyzing the problem, writing conditions in order to solve those problems. Followed by evaluating the program, debugging of issues and referring to the problem to fix errors.

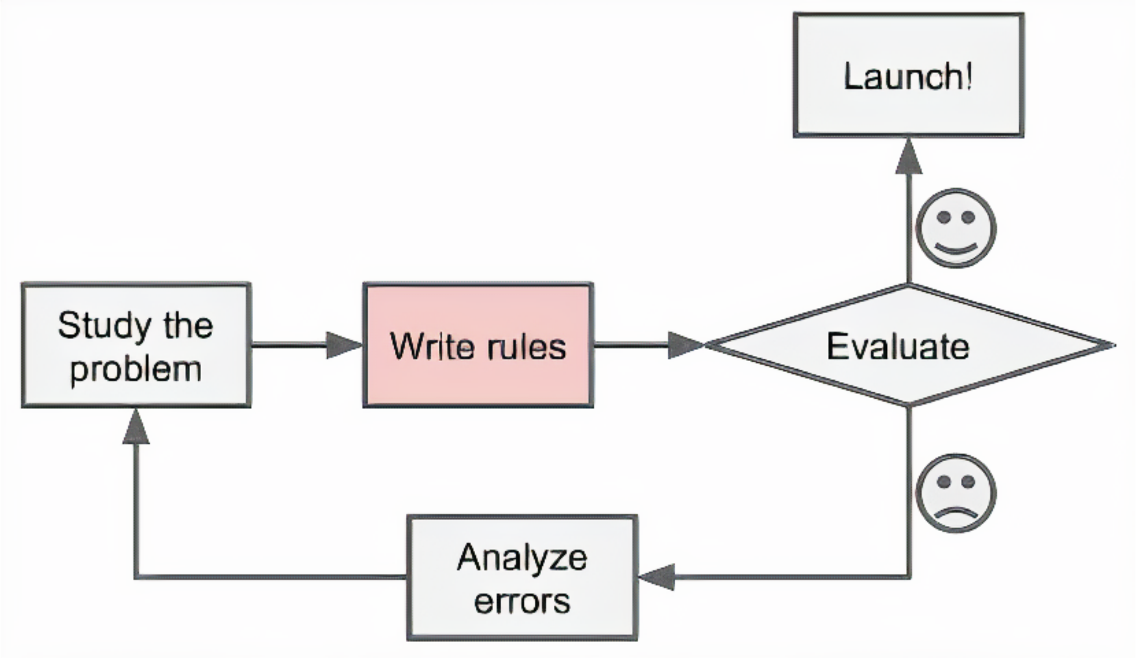


Figure 10. *Traditional Approach of programming [14]*

Oppositely, Machine Learning is providing new approach of solving problems. The machine learning approach is by guiding machine on correct result and delegating technique to itself.

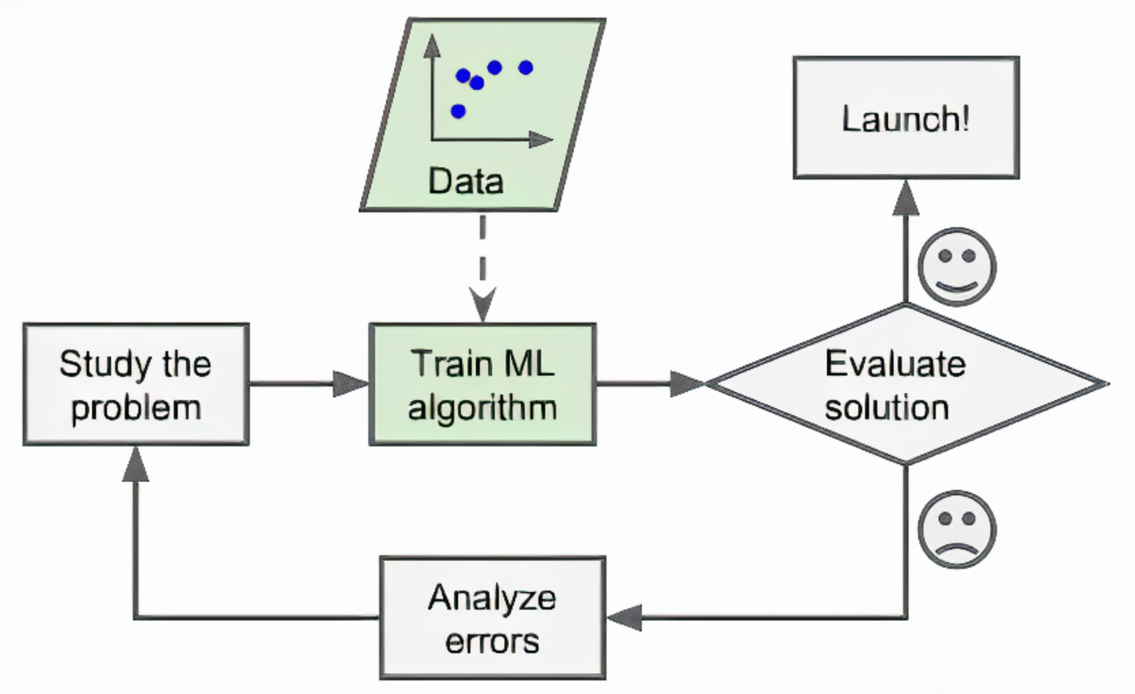


Figure 11. *Machine Learning approach [14]*

2.1.1 Machine Learning

There are three major branches of Machine Learning which are Supervised learning. Unsupervised learning and Reinforcement learning. In this section, supervised and unsupervised learning will be discussed.

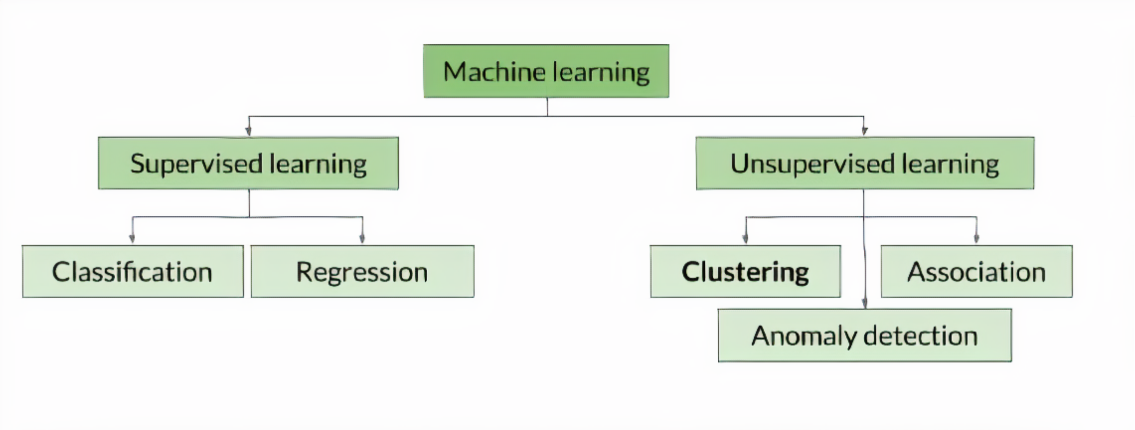


Figure 12. *Machine Learning Tree and branches [15]*

Supervised Learning

Supervised learning is the machine learning where output is generated by giving labeled training data as an examples. Supervised learning itself has two categories of algorithms of Classification and Regression.

Classification as the name indicates, is a type of supervised learning which solves problems related to classifications. For instance, is there a dog in picture or not?

Below, you can find the workflow of Machine Learning process applicable for supervised learning, The workflow consist of getting raw data, extracting features or labelling the data followed by creating data for training and data for testing the model. The next step is to train the model which uses the data for training. Followed by evaluating of your data with test data. The process of evaluation and returning to training the model is incremental until the model is learned enough to meet your expectations.

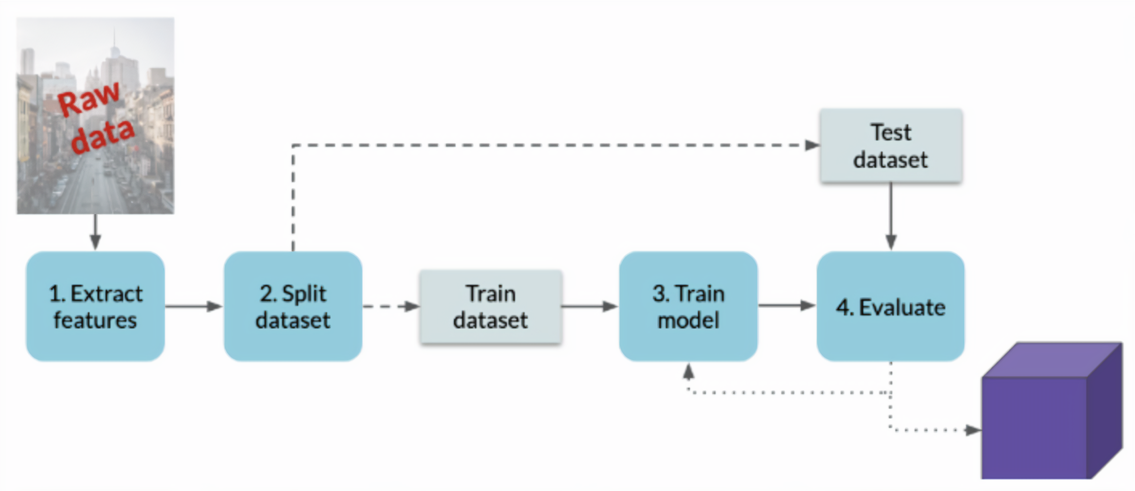


Figure 13. *Workflow of Machine Learning process [15]*

Regression is a category of supervised learning used to solve quantitative problems like predicting age. In regression problem the output is continuous values like price of houses.

Unsupervised Learning

Unsupervised learning is a machine learning category, the model is not trained and it deals with unlabelled data. The applications of unsupervised learning is very significant in solving very complex tasks very efficiently.

There following applications of unsupervised learning:

- Clustering. Clustering is the process of dividing the data and building groups of different clusters.
- Visualization. Creating diagrams, graphs and charts in order to illustrate some

information is called visualization. Unsupervised learning is used for this method.

- Dimensionality Reduction. The process of decreasing entropy of variables is called dimensionality reduction. Basically, the aim is to make data simple without losing information.
- Finding Association Rules. The process of finding connection between different values in data is called finding associations. For example, measuring the probability that if some persons likes X type of videos on YouTube he will like Y type of videos too.
- Anomaly Detection.As the name implies, is finding items which is different from the normal data.

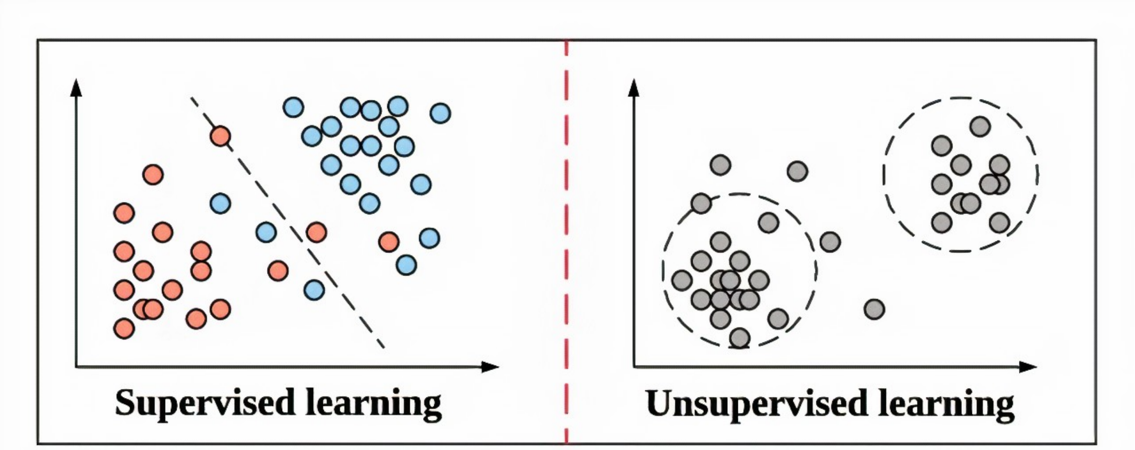


Figure 14. *Examples of Supervised and Unsupervised Learning [14]*

2.1.2 Neural Networks

Neural networks are a set of algorithms inspired by the human brain and designed to recognize patterns. They interpret the data through based on labeling or clustering raw input. Neural Networks are constructed of processing units called as neurons or nodes, which looks like how biological neurons work as illustrated below.

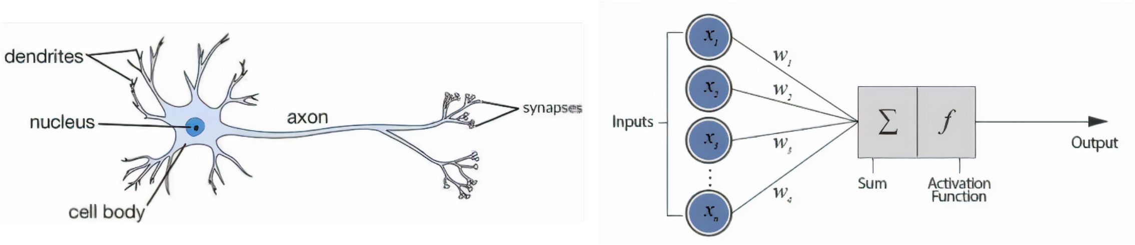


Figure 15. *Biological Neuron and Artificial Neuron [13]*

In the figure below you can find the example of Neural Network called Perceptron, which is single layer neural network. Perceptron network accepts a number of inputs, make computation of a weighted sum, and generates a step function for the final prediction. It is usually used in order to make binary classification, like dog or not dog problem types.

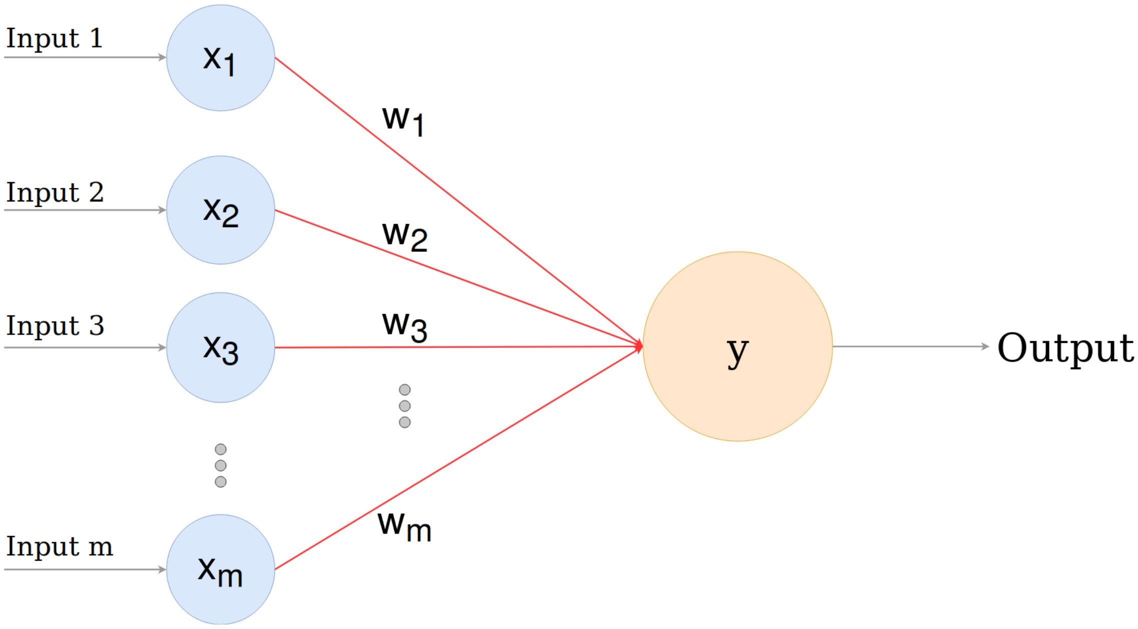


Figure 16. *Perceptron structure [16]*

In other words, the step function other know as activation function is a weighted sum of inputs:

$$\sum_{i=1}^{n}(W_i * X_i) = a$$

Whereas Neural networks consist of multiple layers namely input, hidden and output layers. The input layer or visible layer represents the data in the raw form. For facial emotional recognition models it will be labelled photos of representing different emotions. The hidden layer helps the neural network to develop new representations of the input data which is used to learn complex relationships between the data and the labels.

Finally, the output layer is the last layer in the model returning the output in the case of facial emotional analysis it is one of basic emotions [17].

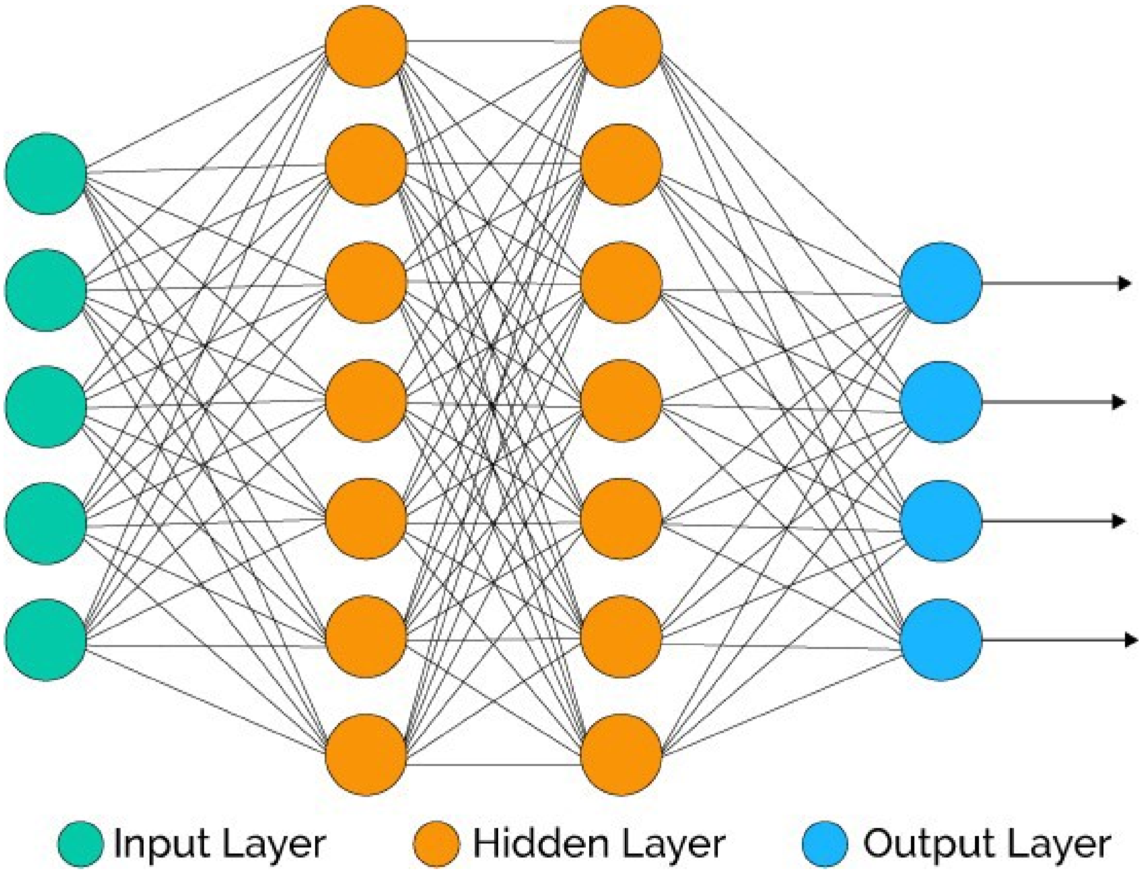


Figure 17. *Neural Networks Structure [17]*

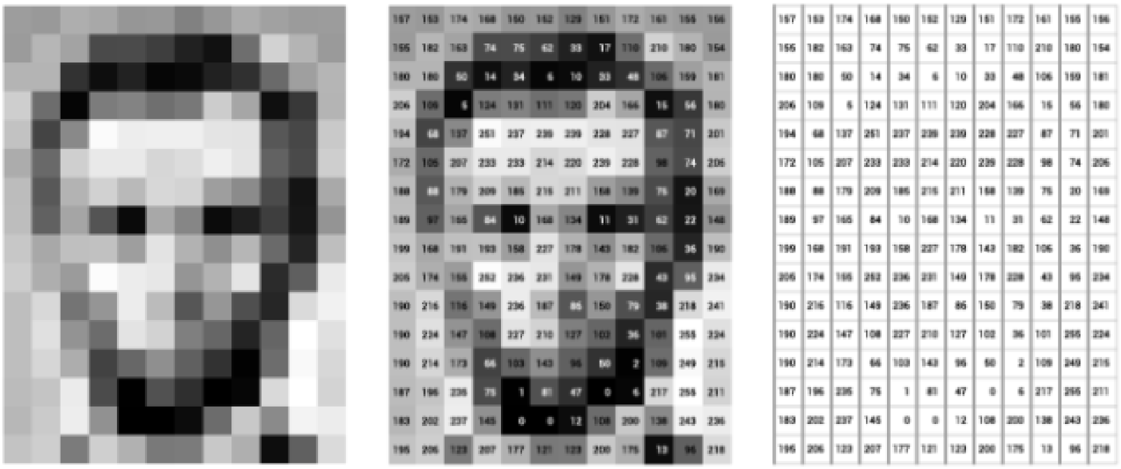
2.2 Fundamentals of Image Processing

In this subsection, the fundamentals of image processing will be discussed, specifically how computers perceives images and how to perform operations on images.

Every Image is 2D representation of 3D objects represented in the form of a grid of pixels and Every pixel in an image is just a numerical value. For example, in the gray scale range the pixels of image will have values from 0 to 255 with 0 being black and 255 being white. It is possible to change those pixel values and perform operations on them. can also change these pixel values. We can multiply every single one by a scalar to change how bright the image is, we can shift each pixel value to the right, and many more operations.

Like from image below it is clear that every picture composed of pixels and every pixels has value from 0 to 255 representing darkness.

In addition, every image is represented by resolution which is dimension of image, representing as the number of image elements per measurements like dots per inch [18].



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Figure 18. *Image Data [18]*

Image Coordinate System

In image processing, each square in the coordinate plane is known as a pixel and hold a value acquired after capturing and image.

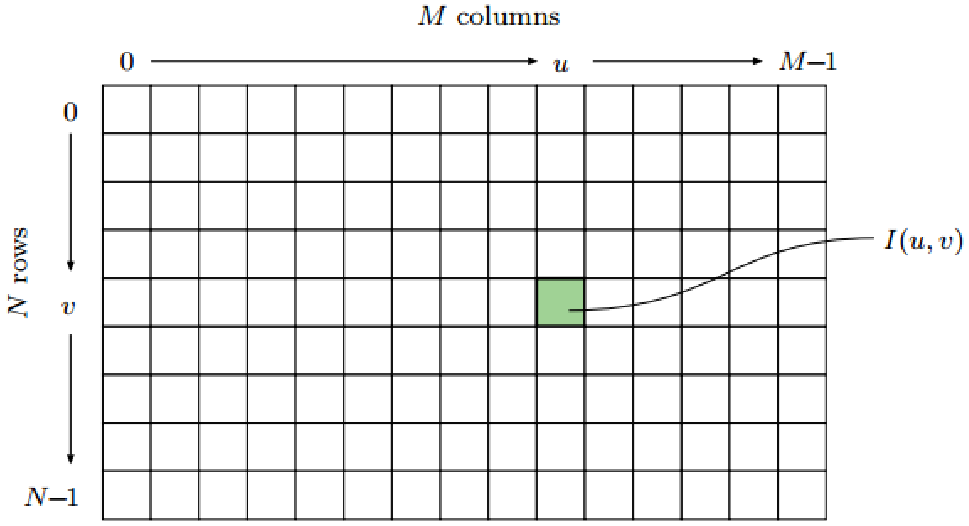


Figure 19. *Image Data [18]*

2.2.1 Colored Images

On other hand every colorful image could be represented by as 3D cubes of values with width, height, and depth and 3 color channels. Channels basically are the number of colors in an image.

Below you can see image of Lenna or Lena, which is standard test image used in image processing field.

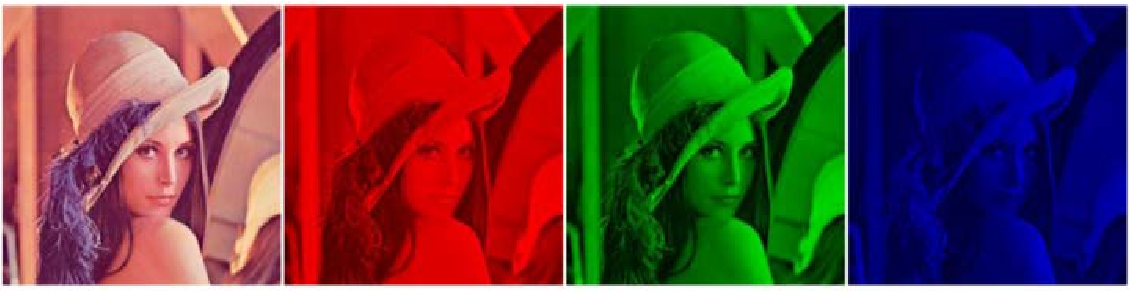


Figure 20. *Decomposition method of RGB images. (1) Original Lena image; (2) Red component; (3) Green component; (4) Blue component [19]*

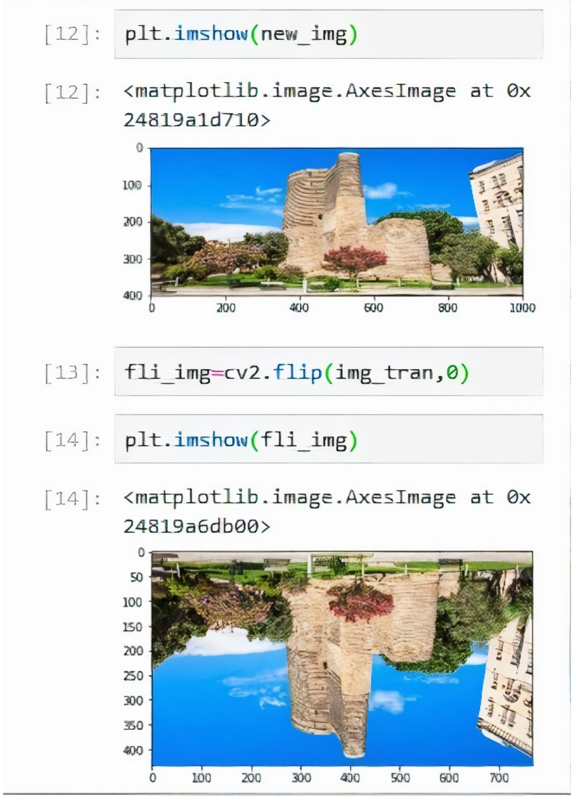
The depth is the number of colors and color images can be represented by combinations of 3 colors: red, green, and blue values. These are known as RGB images. Dept of RGB imnage is respectively 3. In other words, every colorful image is composed of 3 layers: one layer is Red, one Green, and one Blue. Together they create a complete color image.

In general, the necessity of all 3 color channels is vary and determined depending on task. For instance, if autonomous car project is done then all 3 colors channels are needed, because road signs and traffic light are colorful. On other hand, for facial emotional recognition, not all 3 color channels are needed.

2.2.2 Operations on Images with Python Image Library

In addition to basics, the it is possible to perform various operations on images.

For example, like in image of Maiden Tower of Baku, it is possible to resize and rotate any images. It is possible to assign a size by giving a variable newsize = (300, 300) and resize it with following code snippet im1 = im1.resize(newsize) [20].

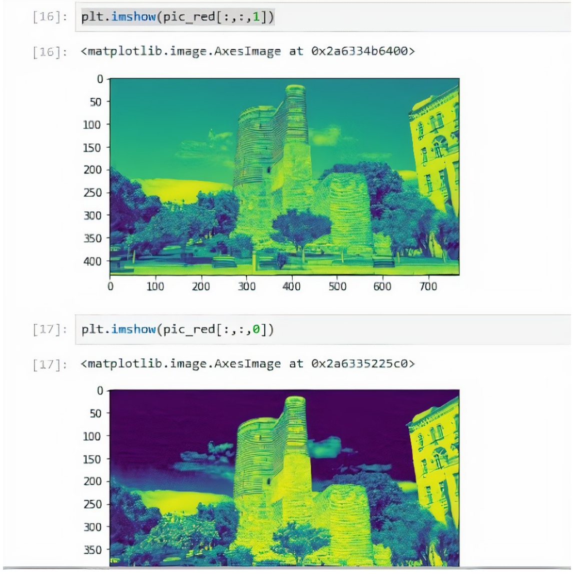


```
[12]: plt.imshow(new_img)

[12]: <matplotlib.image.AxesImage at 0x
      24819a1d710>
```

```
[13]: fli_img=cv2.flip(img_tran,0)

[14]: plt.imshow(fli_img)

[14]: <matplotlib.image.AxesImage at 0x
      24819a6db00>
```

Figure 21. *Python Imaging Library operations [21]*

In addition, it is possible to remove color channels from images with following code snippet plt.imshow(pic[: , : , 0]). Last digit in code above which is 0 means Red channel, 1 is Green channel and 2 is Blue channel. [20]

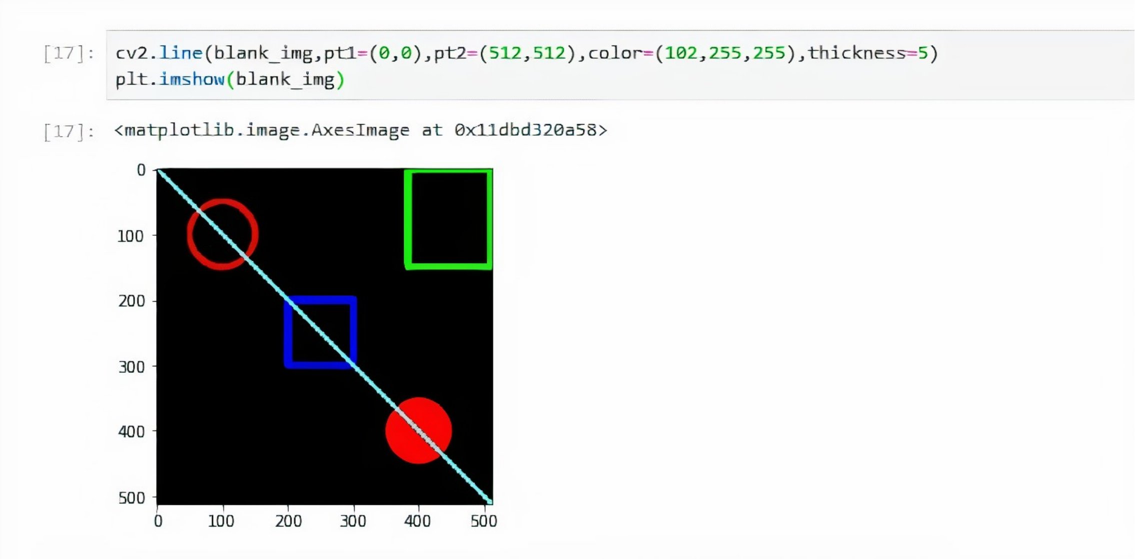


```
[16]: plt.imshow(pic_red[:,:,1])

[16]: <matplotlib.image.AxesImage at 0x2a6334b6400>
```

```
[17]: plt.imshow(pic_red[:,:,0])

[17]: <matplotlib.image.AxesImage at 0x2a6335225c0>
```

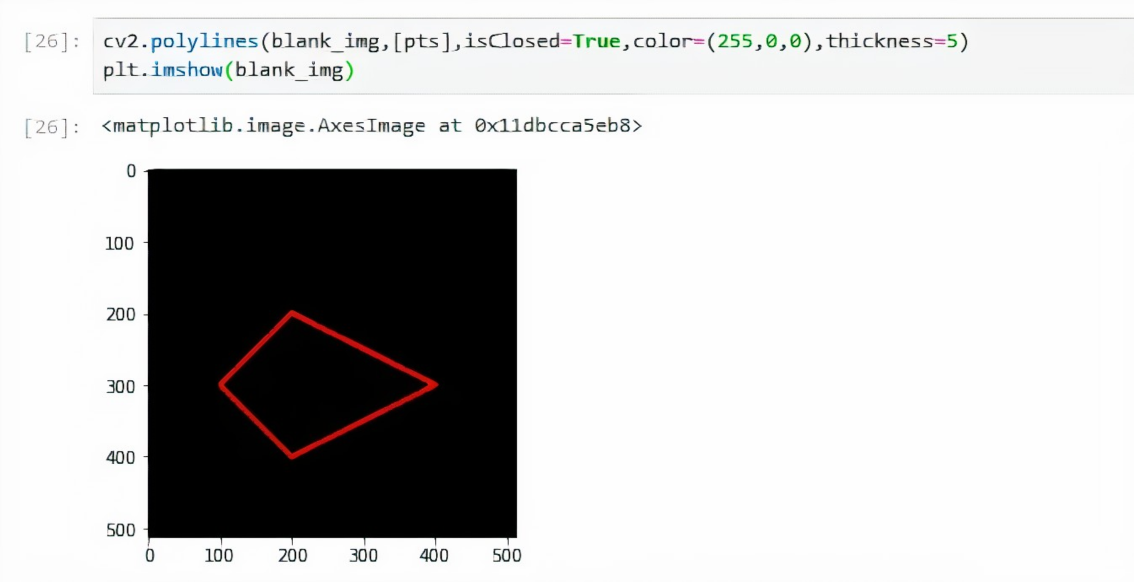
Figure 22. *Python Imaging Library operations [21]*

Apart from that, it is possible to draw various shapes on images, lines, circles, triangles, rectangles and even poly lines like in images below.



```
[17]: cv2.line(blank_img,pt1=(0,0),pt2=(512,512),color=(102,255,255),thickness=5)
      plt.imshow(blank_img)

[17]: <matplotlib.image.AxesImage at 0x11dbd320a58>
```

Figure 23. *Python Imaging Library operations [21]*



```
[26]: cv2.polylines(blank_img,[pts],isClosed=True,color=(255,0,0),thickness=5)
      plt.imshow(blank_img)

[26]: <matplotlib.image.AxesImage at 0x11dbcca5eb8>
```

Figure 24. *Python Imaging Library operations [21]*

2.3 Software Dependencies

In this section, software dependencies and importance to use them in project are described

2.3.1 Python

Python is a open-source, general purpose and high-level scripting language with relative smooth learning curve and with strong libraries for machine learning.

Many operating systems including Raspbian comes with default python and it's packages. In total python has thousands of third-party libraries, including scientific libraries. In this chapter we will discuss, libraries which are used in research.

2.3.2 Virtual Environment

The Venv is a virutal environment module to create virtual environments with their own directories and dependencies. It helps to easily troubleshoot errors and to have modularity.

There is no need to install venv because it comes with python installation. You need to run following command in order to initialize a venv project [22]:

```
python3 -m venv projectName
```

It's needed to activate the project in order to start using virtual environment by command below:

```
source projectName/bin/activate
```

After activating the project it is easy to install packages like this example:

```
pip install requests
```

Also running the script:

```
python main.py
```

deactivate command is used to deactivate virtual environment.

2.3.3 Numpy

Numpy is scientific library used with python programming language. The main usage of this library include manipulating and using n-dimensional arrays. Operations on multi-dimensional arrays could be performed with the help of this library. Following command should be used to install numpy.

```
pip install numpy
```

2.3.4 Pandas

Pandas is another open-source library used for data analysis with Python programming language. It helps to read and write data with many different formats, reshaping and pivoting data sets. Following command should be used to install numpy.

```
pip install pandas
```

2.3.5 Tensorflow

TensorFlow is a open-source software library, developed by Google used for machine learning applications such as neural networks.

It is possible to create dataflow graphs—structures that describe how data moves through a graph, or a series of processing nodes. Every node is representing some machine learning model.

Both CPU and GPU are used to perform computations in TensorFlow.

2.3.6 Tensorflow Lite

TensorFlow Lite as a name tells is a lite version of Tensorflow, and it is a set of tools to run TensorFlow models specifically on mobile, embedded, and IoT devices. Machine Learning inference with low latency and a small binary size is done.

Basically, The TensorFlow Lite converter takes a TensorFlow model and generates a file (.tflite) called as FlatBuffer. This file is then could be deployed to the client device like mobile or embedded in our case.

It increases efficiency by helping to train model in one device like PC or cloud and then making inference in other device like embedded system as it visible from conversion process in the figure below:

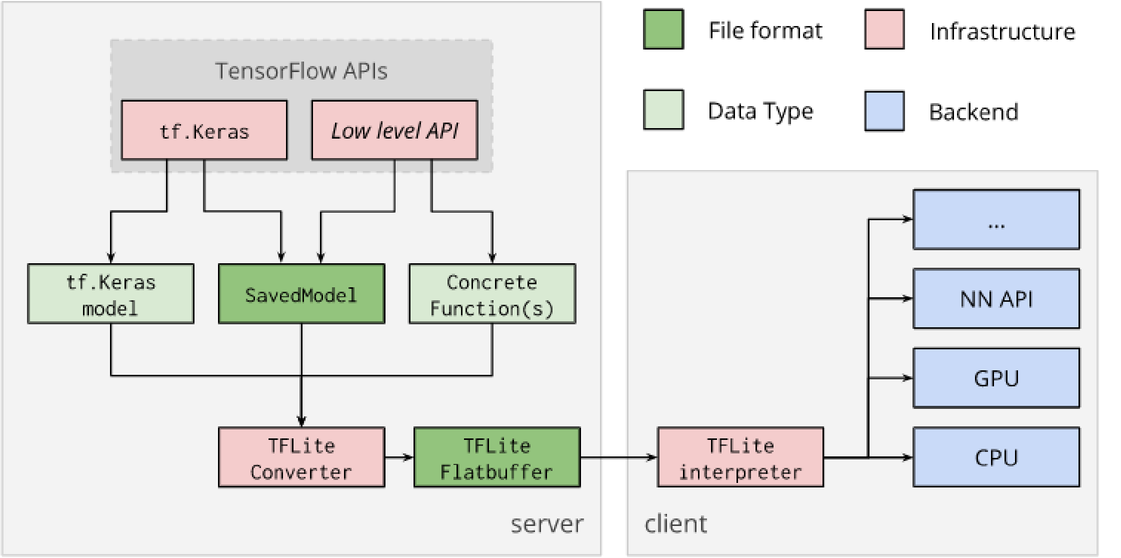


Figure 25. *Tensorflow Lite Architecture [23]*

2.4 Conclusion

In summary, there are many advancements in the area of computer vision and deep learning. Every machine learning model goes trough a workflow consisting of data collection, data cleaning, training the model and testing the model with test data, and this process goes until reaching desired accuracy.

Two features are needed to train a good model which are enough amount of Image data sets, and requires good enough computing resources. However, it is possible to use Tensorflow Lite to move the model to mobile or embedded device. Thus, removes necessity to train model on the System on Chip and rely only inference of classification model on embedded device.

In addition, this chapter demonstrated that there are many libraries which help to easily manipulate images, train a neural network model and even apply those models to embedded systems.

3. Methodology and Design

This chapter we will describe the pipeline of facial emotional analysis 's that are used in implementation of experiments.

3.1 Pipeline of Facial Emotional Analysis

First of all, a computer receives and input from camera and pre-processing happens. Pre-processing is done in order to make standardisation of image by lowering the quality, or changing from color to gray scale. Next the images are analysed and face detection is performed, followed by finding facial markers of mouth, eyes, cheek, etc. Then it is fed to the trained model which performs recognition and prediction of some probability of having specific emotion.

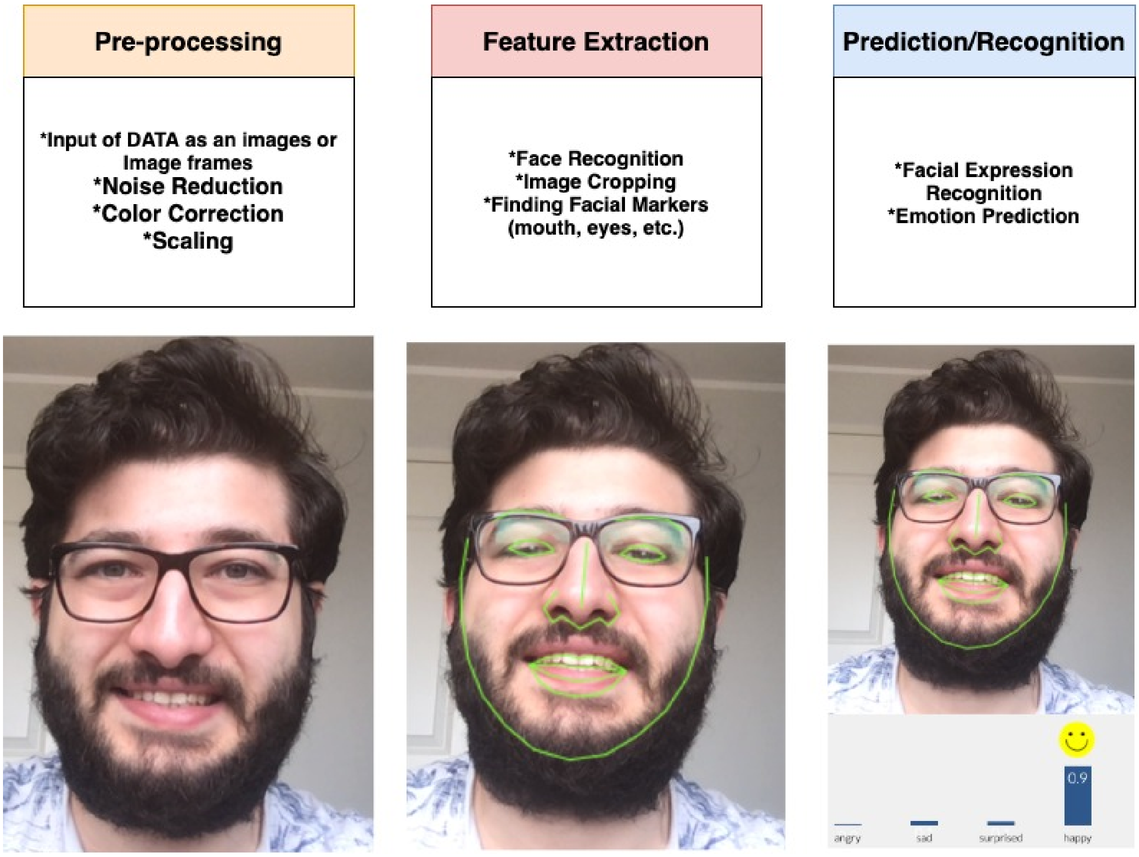


Figure 26. *Pipeline of facial emotional recogntion [17]*

3.1.1 Face Detection

Face Detection should be performed before doing facial emotional analysis. The Face Detection algorithms are described in this subsection.

Viola-Jones Algorithm

Viola-Jones, invented by Paula Viola and Michael Jones in the early 2000s, uses decision trees to detect faces based on light and dark areas. Not very accurate but very fast and suitable for low-powered devices. The algorithms gives low false positives but higher true positives [24].

In general it consists of 4 steps namely Haar Features detection, Creating an Integral Image, Adaboost Training and Cascading classifiers.

A Haar-like feature is adjacent rectangular regions in some specific location, named as detection window which sums up the pixel intensities in each region and calculates the difference between these sums.

The Five Haar-like patterns are displayed in figure below. In the figure below, Haar Like Features are displayed.

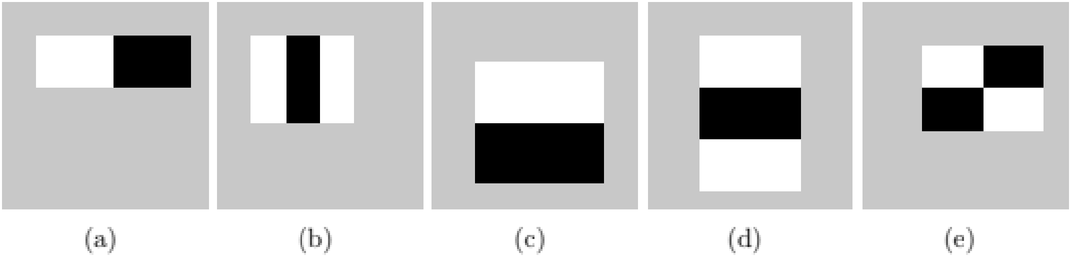


Figure 27. *Haar-Like Features [25]*

Face features like eyebrow, eye, nose and lips could be found by applying those Haar-Like features, like illustrated in image below.

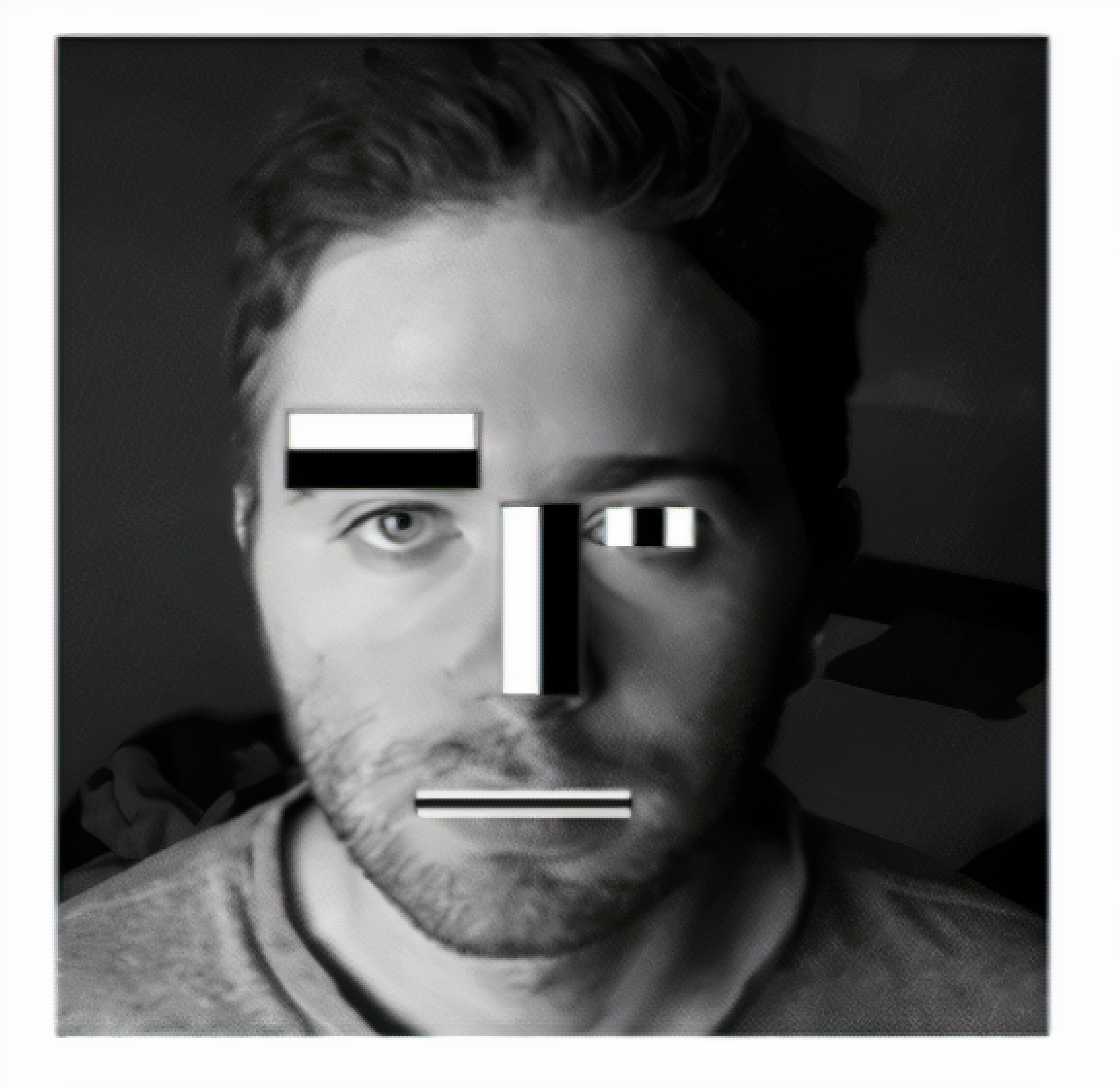


Figure 28. *Haar-Like Features [26]*

Face Features are identified with the help of the rectangles representing Haar Features. The integral image is generated from the sum of the values of the grid.

Afterwards, we apply Adaboost learning algorithm in order to train our classifiers. It constructs strong classifier from weaker classifiers. The idea is that each classifier alpha might be weak by itself but when all that classifiers are applied together, it strengthens the classifier function.

$$F(x) = \alpha_1 * f_1(x) + \alpha_2 * f_2(x) + \alpha_3 * f_3(x) + ... = \sum_{i=1}^{n} (\alpha_i * f_i(x) \tag{3.1}$$

The training classifier or Haar Cascade Classifier is used to detect the face or faces in the image, by using the sum of weak classifiers which builds stronger classifier.

3.1.2 Active Shape Model

The next step after detecting face in image is to extract facial points by using Active Shape Model algorithm.

Active Shape Models are statistical models which deforms by iterations to fit the shape of object in a new image. That shape of an object is represented as set of point controlled by the shape of model. The algorithm aims to match the model to a new image.

Points which could be located from one image to another should be considered for landmarks. The easiest way to generate a training set is to label image with related points which in practice is time consuming. In practice, automatic and semi-automatic methods are used.

In two dimensional images, the points could be placed in the corners of object boundaries, the junctions in the form of T letter can be easily located as biological landmarks.

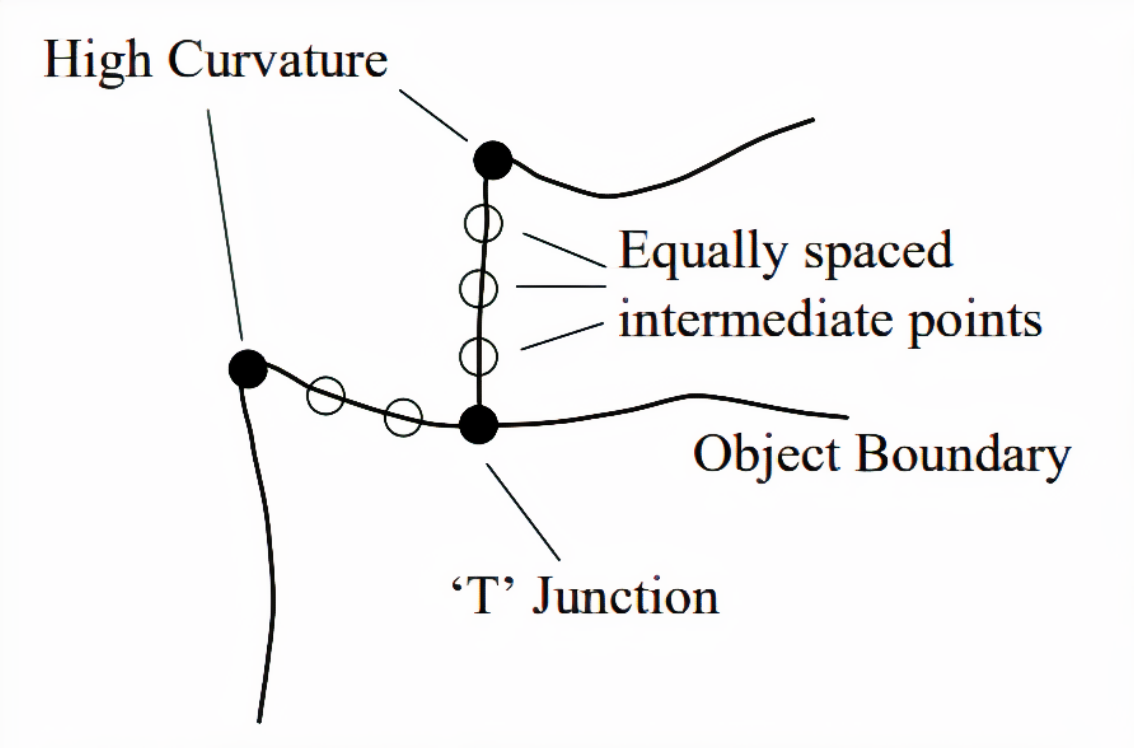


Figure 29. *: Good landmarks are points of high curvature or junctions [27]*

If a shape is represented with n points in d dimensions we form the shape by a nd element vector formed by adding the elements of the individual point position vectors. In a 2-D image we can represent the n landmark points as a set of

$$(x_i, x_i) \tag{3.2}$$

where

$$x = (x_1, ..., x_n, y_1, ..., y_n) \tag{3.3}$$

Below you can see an example how Active Shape Model Search progressing with every iteration.

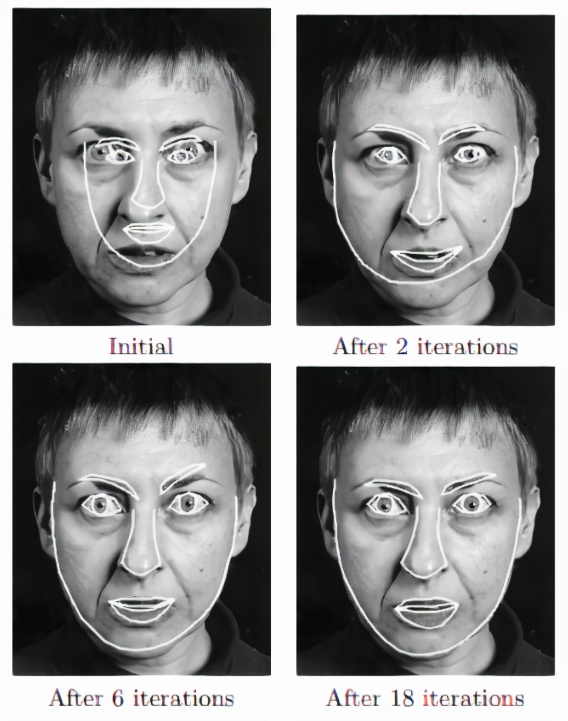


Figure 30. *Active Shape Model Search progressing [27]*

3.1.3 Facial Emotional Recognition with Convolutional Neural Networks and AdaBoost

AdaBoost is a machine leaning algorithm which uses sum of small weak classifiers in order to form a bigger and stronger classifier.

Classifiers are built based on features that differentiate emotions based on muscles involved in expressing those emotions. The facial muscles involved in expressing emotions are well defined and universal like it is visible from figure below:

Emotion	Darwin’s Facial Description
Fear	Eyes open, Mouth open, Lips retracted, Eyebrows raised
Anger	Eyes wide open, Mouth Compressed, Nostrils raised
Disgust	Mouth open, Lower lip down, Upper lip raised
Contempt	Turn away eyes, Upper lip raised, Lip protrusion, Nose wrinkle
Happiness	Eyes sparkle, Mouth drawn back at corners, Skin under eyes wrinkled
Surprise	Eyes open, Mouth open, Eyebrows raised, Lips protruded
Sadness	Corner of mouth depressed, Inner corner of eyebrows raised
Joy	Upper lip raised, Nose labial fold formed, Orbicularis, Zygomatic

Figure 31. *Description of Facial Muscles involved in expressing emotions [28]*

The performance of Convolutional Neural Network with AdaBoost is improved by every iteration of epochs.

Training Dataset

Training dataset is provided by Kaggle website, which has of about 37 thousands of 48*48 pixel gray-scale images of faces. The face in images are almost centered and face has the same amount of space in every image. Each image has to be categorized into one of the seven classes namely as: 0 for Angry, 1 for Disgust, 2 for Fear, 3 for Happy, 4 for Sad, 5 for Surprise, and 6 for Neutral.

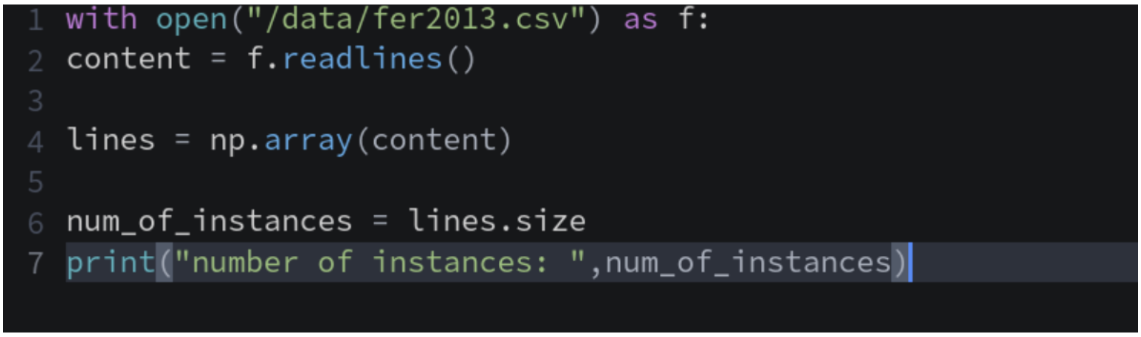
3.2 Raspberry Pi Development

In this section, Raspberry Pi development is described including building and training the model in desktop, configuring Raspberry Pi and uploading model to Raspberry Pi, and performing model inference. Code and configurations are described in following section.

3.2.1 Training Model

Different parts of Fec2013 dataset is used for training the model. Compressed version of dataset takes 92 MB space, uncompressed takes 295 MB space. There are 28K training and 3K testing images in dataset. Each image has 48x48 pixel matrix. Pure data set consists of image pixels (48x48=2304 values each), emotion of each image and usage type.

Dataset already has loaded under data folder. The way of reading dataset is below:

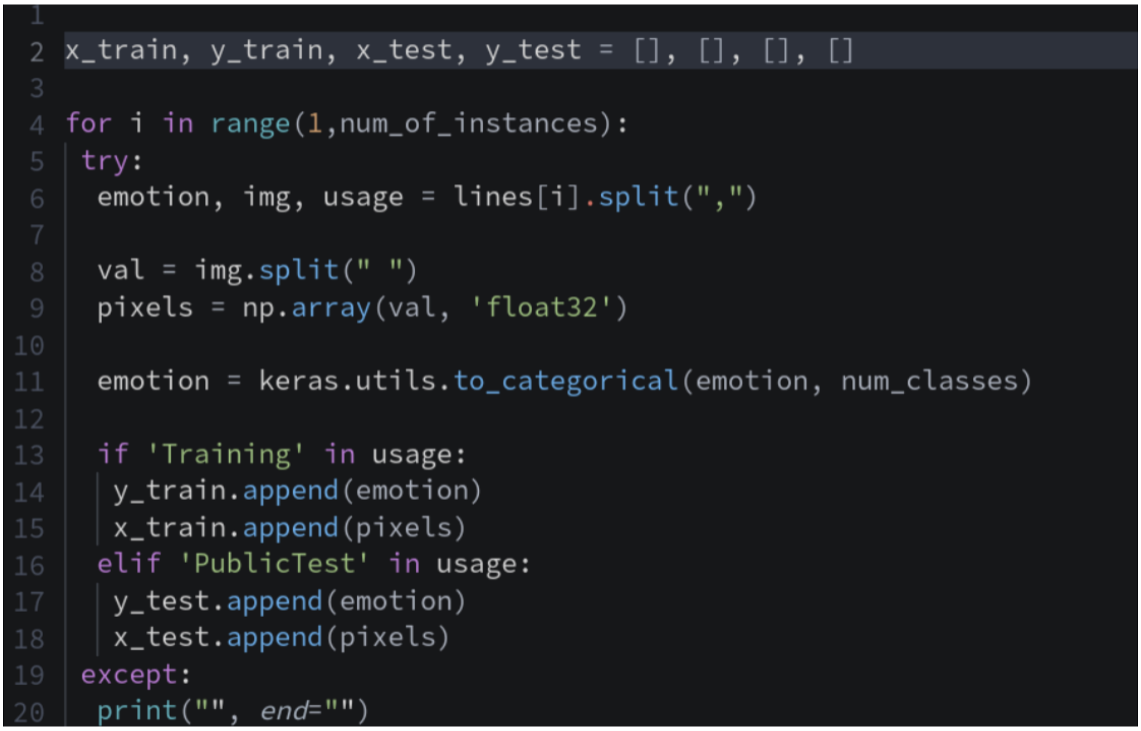


```
with open("/data/fer2013.csv") as f:
content = f.readlines()

lines = np.array(content)

num_of_instances = lines.size
print("number of instances: ",num_of_instances)
```

Figure 32. *Accessing Dataset*

Learning Procedure

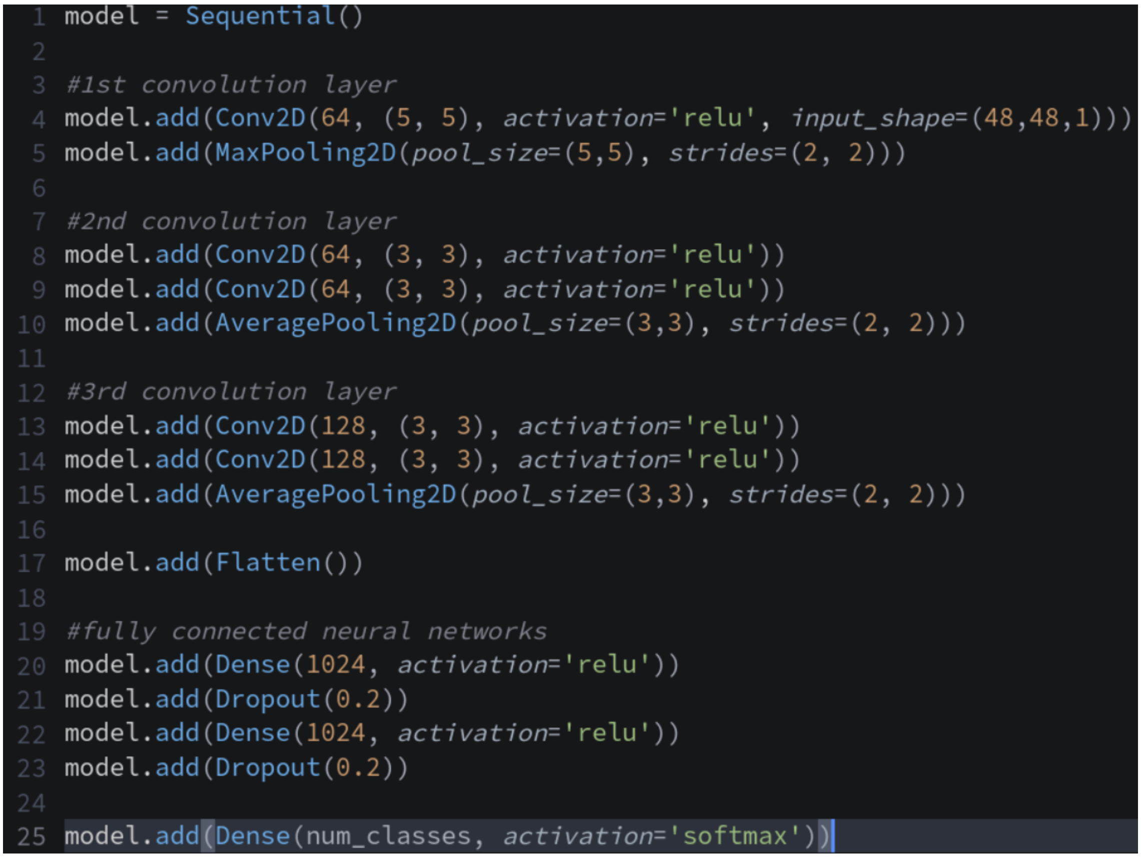
Using Deep Learning, convolutional neural networks applied to tackle this task and constructed CNN with Keras using TensorFlow backend. Train and test set can be stored into dedicated variables.



```

x_train, y_train, x_test, y_test = [], [], [], []

for i in range(1,num_of_instances):
 try:
  emotion, img, usage = lines[i].split(",")

  val = img.split(" ")
  pixels = np.array(val, 'float32')

  emotion = keras.utils.to_categorical(emotion, num_classes)

  if 'Training' in usage:
   y_train.append(emotion)
   x_train.append(pixels)
  elif 'PublicTest' in usage:
   y_test.append(emotion)
   x_test.append(pixels)
 except:
  print("", end="")
```

Figure 33. *Training Model with Deep Learning*

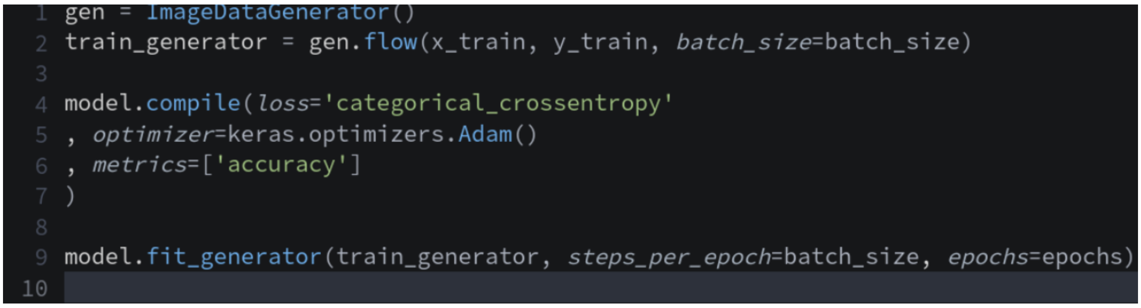
The structure of CNN is constructed below:



```
model = Sequential()

#1st convolution layer
model.add(Conv2D(64, (5, 5), activation='relu', input_shape=(48,48,1)))
model.add(MaxPooling2D(pool_size=(5,5), strides=(2, 2)))

#2nd convolution layer
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(AveragePooling2D(pool_size=(3,3), strides=(2, 2)))

#3rd convolution layer
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(AveragePooling2D(pool_size=(3,3), strides=(2, 2)))

model.add(Flatten())

#fully connected neural networks
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(num_classes, activation='softmax'))
```

Figure 34. *CNN structure*

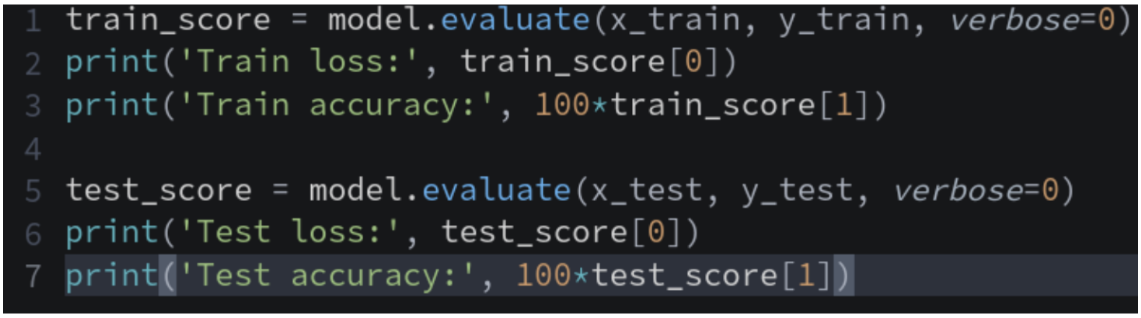
For quick completion, implement learning with randomly selected train set instances is useful. That is the reason why train and fit generator used.



```
gen = ImageDataGenerator()
train_generator = gen.flow(x_train, y_train, batch_size=batch_size)

model.compile(loss='categorical_crossentropy'
, optimizer=keras.optimizers.Adam()
, metrics=['accuracy']
)

model.fit_generator(train_generator, steps_per_epoch=batch_size, epochs=epochs)
```

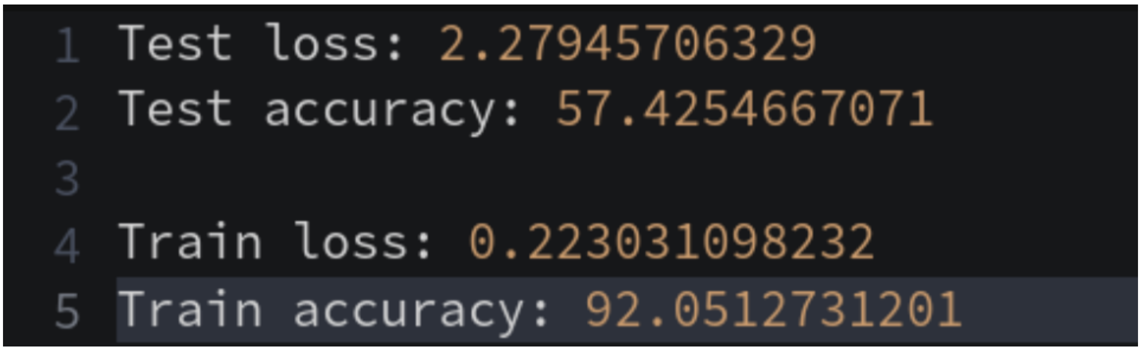
Figure 35. *Training Model with Deep Learning*

Following code below performs evaluation of model.



```
train_score = model.evaluate(x_train, y_train, verbose=0)
print('Train loss:', train_score[0])
print('Train accuracy:', 100*train_score[1])

test_score = model.evaluate(x_test, y_test, verbose=0)
print('Test loss:', test_score[0])
print('Test accuracy:', 100*test_score[1])
```

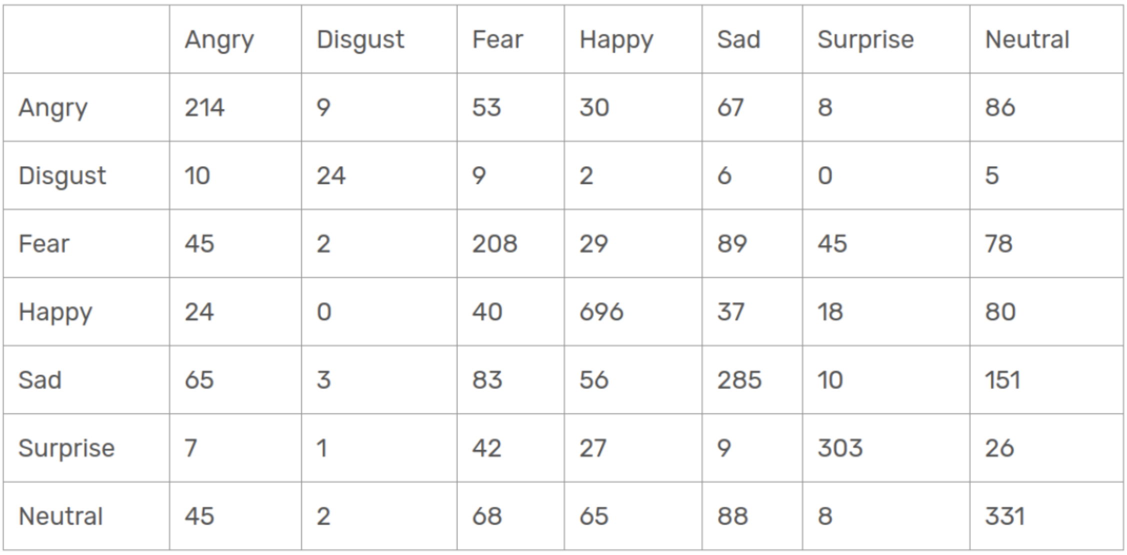
Figure 36. *Evaluation the Model*



```
Test loss: 2.27945706329
Test accuracy: 57.4254667071

Train loss: 0.223031098232
Train accuracy: 92.0512731201
```

Figure 37. *Evaluation the Model*

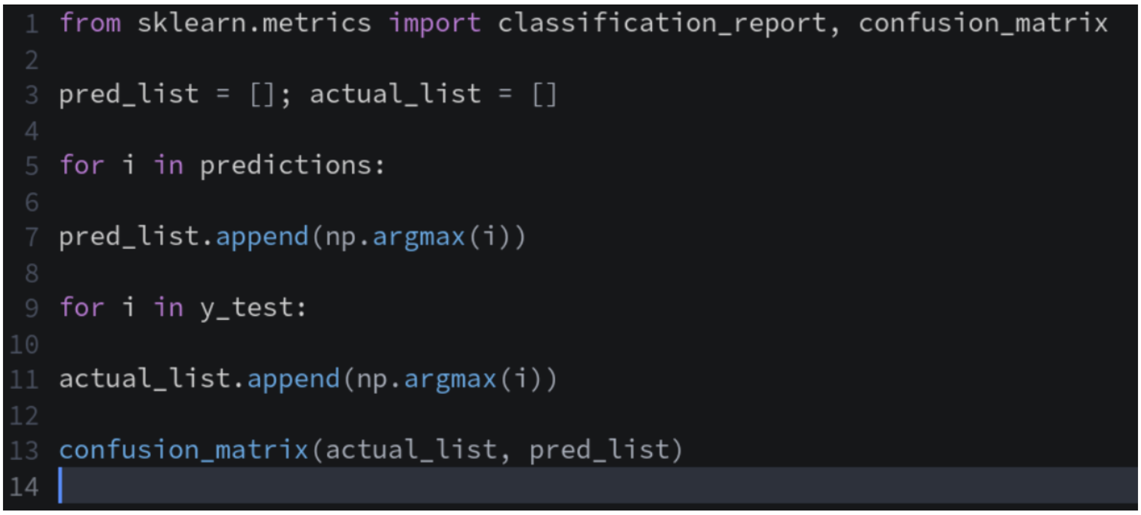
Despite of measuring test accuracy, confusing matrix works better for evaluating deep learning models of classification problems with multiple classes. Confusion matrix of this model is demonstrated below. Lines represent actual values whereas columns state predictions.



	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	214	9	53	30	67	8	86
Disgust	10	24	9	2	6	0	5
Fear	45	2	208	29	89	45	78
Happy	24	0	40	696	37	18	80
Sad	65	3	83	56	285	10	151
Surprise	7	1	42	27	9	303	26
Neutral	45	2	68	65	88	8	331

Figure 38. *Confusion Matrix*

Confusion matrix is constructed automatically with the help of Scikit-learn library below:



```
from sklearn.metrics import classification_report, confusion_matrix

pred_list = []; actual_list = []

for i in predictions:

pred_list.append(np.argmax(i))

for i in y_test:

actual_list.append(np.argmax(i))

confusion_matrix(actual_list, pred_list)

```

Figure 39. *Confusion Matrix*

Finally, the model is trained in laptop and evaluated with generating confusion matrix. In the next subsection, the Raspberry Pi configurations will and transfer of trained model to Raspberry Pis will be described.

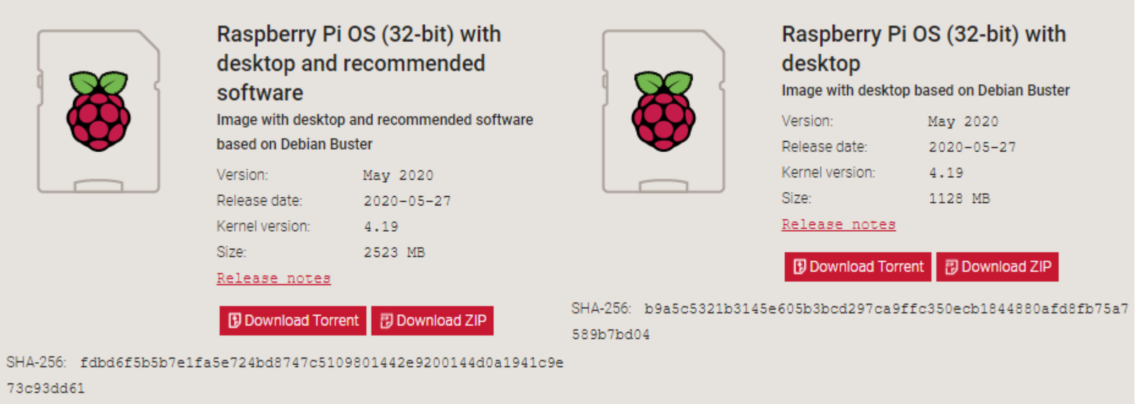
3.2.2 Raspberry Pi Configurations

This subsection will describe necessary configurations in order to run trained facial emotional analysis model on Raspberry Pis.

Installing OS and configuring Headless Connection

Raspberry Pi OS that used to be called Raspbian is the official operating system of Raspberry Pis. In order to install it, we need to download image of Operating System and then burn it with specific tools.

Download one of the Operating Systems below:



Raspberry Pi OS (32-bit) with desktop and recommended software

Image with desktop and recommended software based on Debian Buster

Version:	May 2020
Release date:	2020-05-27
Kernel version:	4.19
Size:	2523 MB

SHA-256: fdbd6f5b5b7e1fa5e724bd8747c5109801442e9200144d0a1941c9e73c93dd61

Raspberry Pi OS (32-bit) with desktop

Image with desktop based on Debian Buster

Version:	May 2020
Release date:	2020-05-27
Kernel version:	4.19
Size:	1128 MB

SHA-256: b9a5c5321b3145e605b3bcd297ca9ffc350ecb1844880afd8fb75a7589b7bd04

Figure 40. *Downloading Operating System's Image [29]*

BalenaEtcher named tools might be used to easily burn the Operating System's image to SD card with one of the specific tools [30].

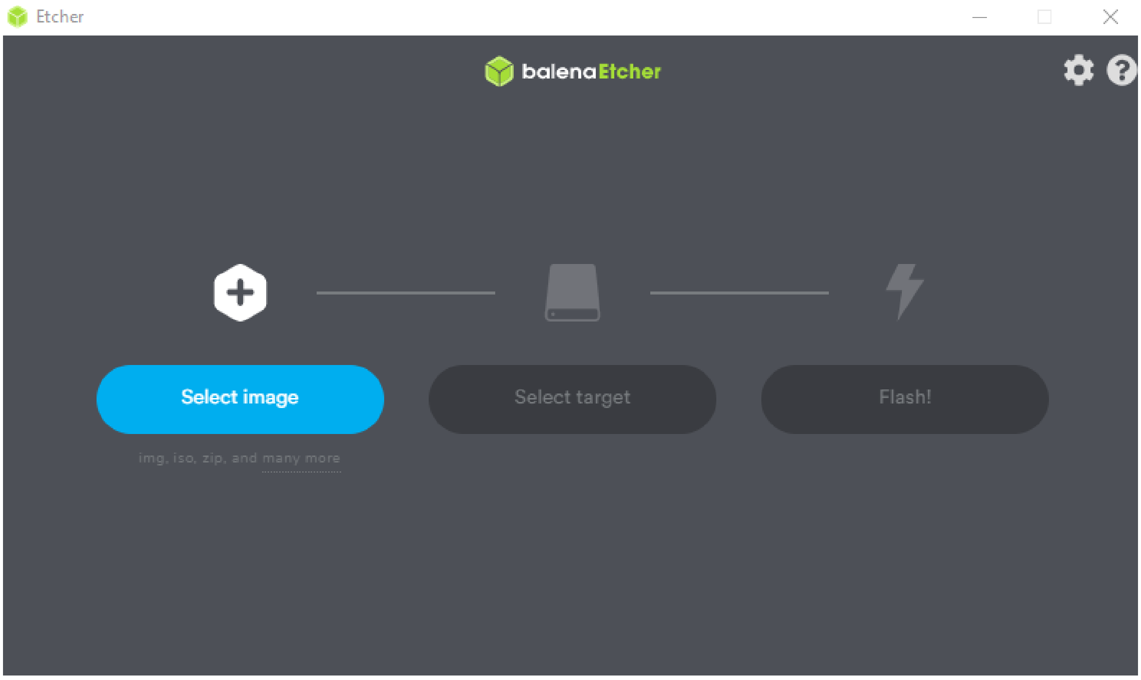


Figure 41. *Burn OS image to micro SD card [30]*

Find the slot underside of Raspberry Pi and insert SD card with operating system. Afterwards, plug it to the power.

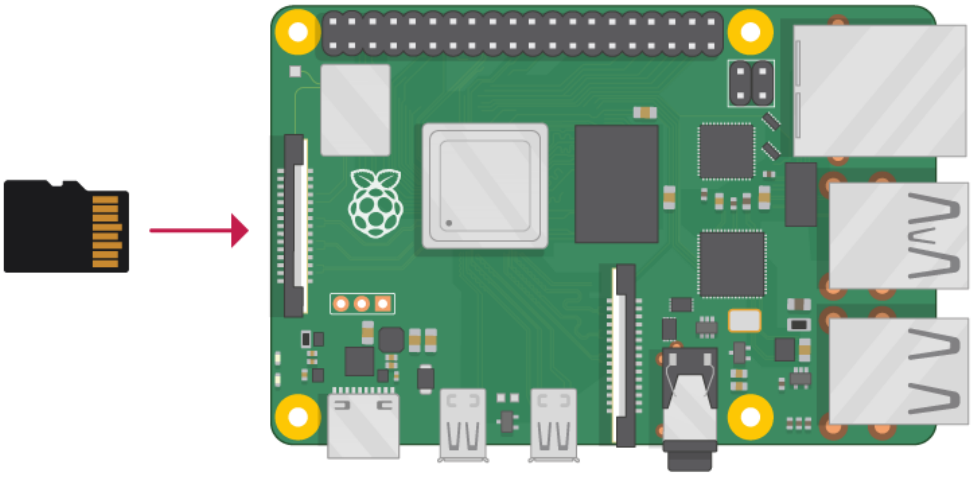
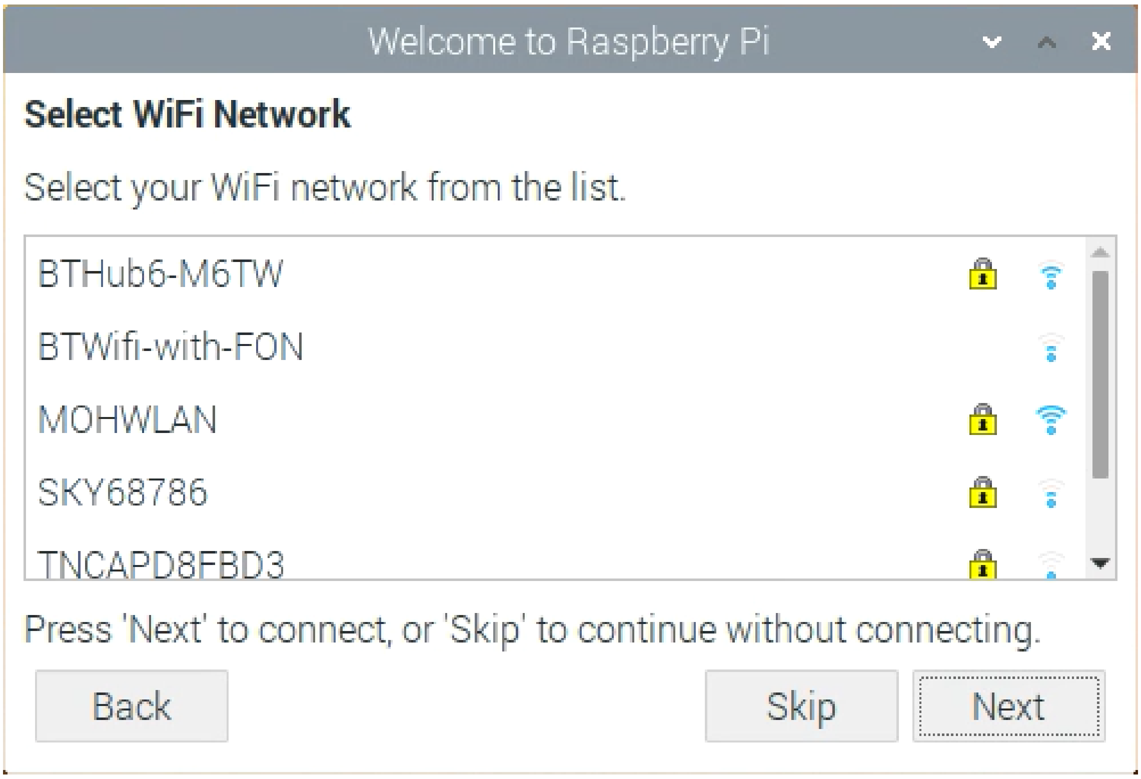


Figure 42. *Connect micro SD card to Raspberry Pi [31]*

After booting Raspberry Pi, configurations of default password for remote connection, timezone, default language and WiFi connection settings could be done.



Welcome to Raspberry Pi

Select WiFi Network

Select your WiFi network from the list.

- BTHub6-M6TW
- BTWifi-with-FON
- MOHWLAN
- SKY68786
- TNCAPD8FBD3

Press 'Next' to connect, or 'Skip' to continue without connecting.

Figure 43. *Configuring WiFi [31]*

Headless Raspberry Pi is needed to eliminate dependency on peripherial devices such as Monitor and keyboard.

VNC viewer software is used to remotely connection and access desktop of Raspberry Pi.

Hotspot WiFi with your work PC and set it as default WiFi node on your Raspberry Pi. Afterwards, find the ip address of connected Raspberry Pi that has VNC interface enabled and put it to VNC Viewer software.

Remote control will be available after connecting to your device with VNC viewer.

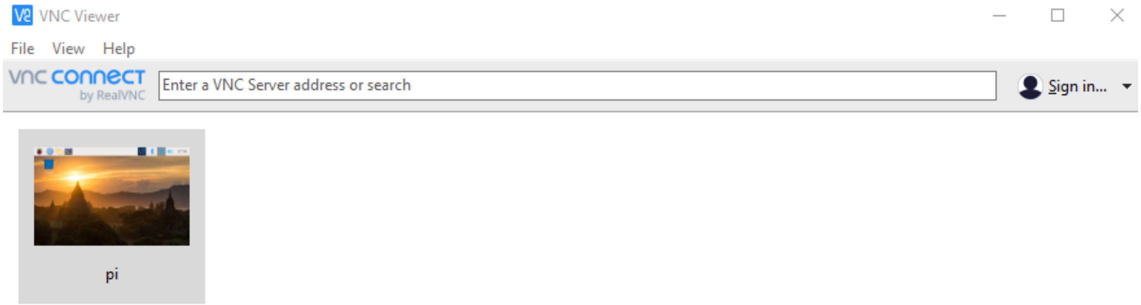


Figure 44. *Use VNC viewer for remote control [32]*

Setting up Raspberry Pi camera

Connecting Raspberry Pi Camera works the same for all types of Raspberry Pis. It's necessary to locate the Camera Module port, pull of the port's plastic clip, insert the Camera Module ribbon cable the right way and push the plastic clip back.

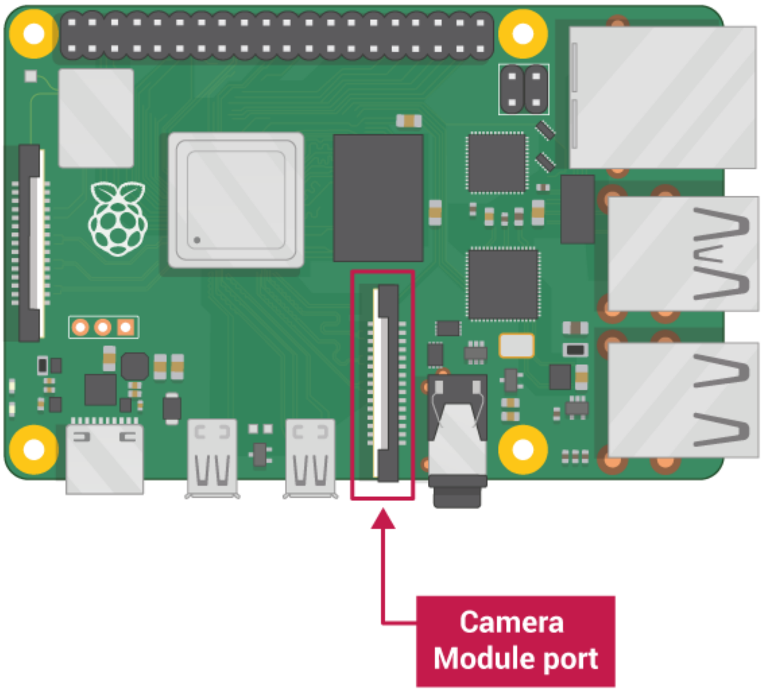
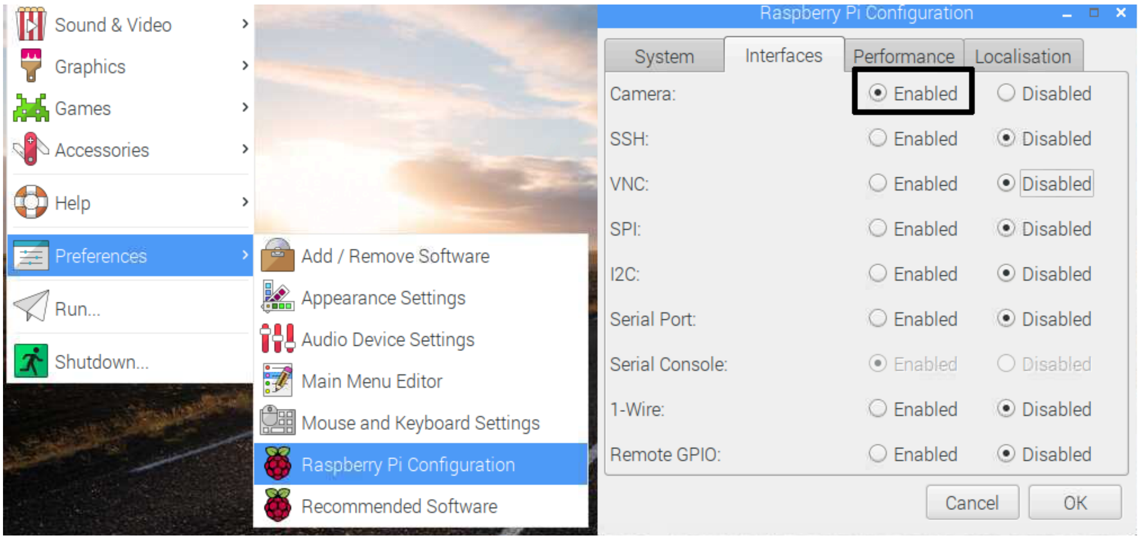


Figure 45. *Pi Camera Module [33]*

In order to be able to use Pi Camera, interface enabling is needed. So, after plugging in Pi Camera Module, switch on Raspberry Pi and enable Camera interface in Interfaces tab of Raspberry Pi Configuration tool. Finally, reboot of Raspberry Pi is needed after saving configurations.



Camera: ● Enabled ○ Disabled

SSH: ○ Enabled ● Disabled

VNC: ○ Enabled ● Disabled

SPI: ○ Enabled ● Disabled

I2C: ○ Enabled ● Disabled

Serial Port: ○ Enabled ● Disabled

Serial Console: ● Enabled ○ Disabled

1-Wire: ○ Enabled ● Disabled

Remote GPIO: ○ Enabled ● Disabled

Figure 46. *Enabling Pi Camera [33]*

Pi Camera should be tested before using it. Taking image or recording video with short commands are a good options. Example command to take a picture with Pi Camera is below:

```
raspistill -o Desktop/image.jpg
```

Video recording is also should be tested for real time analysis. Command below will record a video and save it as video.h264 in Desktop.

```
raspivid -o Desktop/video.h264
```

Setting up USB camera

Considering that Pi Camera has low resolution it's good to use USB streaming cameras as an alternative.

The installation of USB camera is very straightforward with command below [34]:

```
sudo apt install fswebcam
```

Test captures is needed to confirm that web camera is working. Following command will capture an image:

```
fswebcam image.jpg
```

Tensorflow and OpenCV

Tensorflow and OpenCV is needed in order to be able to transfer model from PC and run in Raspberry Pis. Installation of Tensorflow and OpenCV is very straightforward with commands below:

```
pip3 install https://github.com/lhelontra/tensorflow-on-arm/releases/download/v1.14.0-buster/tensorflow-1.14.0-cp37-none-linux_armv7l.whl
```

```
sudo apt-get install python3-opencv
```

3.3 Summary

This chapter address facial emotinal analysis techniques and applies it to raspberry pi microprocessors. Face detection is done by the application of Viola Jones algorithm. Active Shape Models applied to create a landmarks displaying points of faces. Convolutional Neural Networks is trained by Kaggle Facial Emotions dataset.

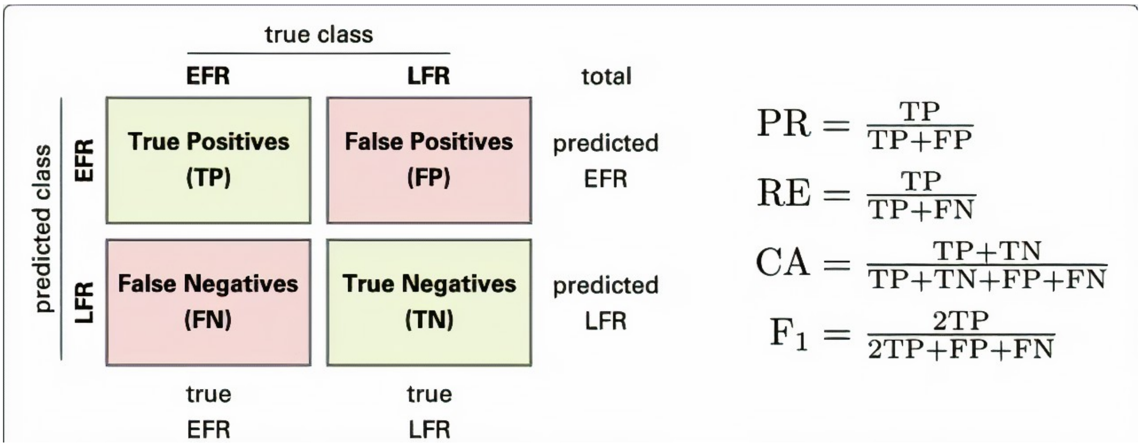
4. Evaluation

In this chapter, the evaluation of neural network models with test data is described and conclusion is made.

Rasberry pi 3 and 4 are evaluated with Confusion Matrix indicating how the model on devices confused while making predictions and by inferencing time which indicates time of execution of model to make classification.

4.1 Confusion Matrix

Confusion Matrix is used In order to give a summary of prediction results. As the name indicates, the confusion matrix shows how the classification model is confused while making predictions by demonstrating the errors being made [35], [36].



		true class		
		EFR	LFR	total
predicted class	EFR	True Positives (TP)	False Positives (FP)	predicted EFR
	LFR	False Negatives (FN)	True Negatives (TN)	predicted LFR
		true EFR	true LFR	

$$\mathrm{PR} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$$

$$\mathrm{RE} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$$

$$\mathrm{CA} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$$

$$\mathrm{F}_1 = \frac{\mathrm{2TP}}{\mathrm{2TP+FP+FN}}$$

Figure 47. *Confusion matrix. Exemplified CM with the formulas of precision (PR), recall (RE), accuracy (CA), and F1-measure [37]*

Accuracy is the number of correct predictions out of all data and it is measured based on percentage of correct predictions by using the test data. It's calculated by dividing of correct predictions by the number of total predictions [38].

$$accuracy = \frac{correctpredictions}{allpredictions}$$

Precision is the percentage of relevant results and it is measured by dividing of all true positive cases to true positive and false positive cases.

$$precision = \frac{truepositives}{truepositives + falsepositives}$$

Recall is the percentage of total relevant results found by algorithm and it is found by dividing true positive examples to the sum of true positives and false negatives.

$$recall = \frac{truepositives}{truepositives + falsenegatives}$$

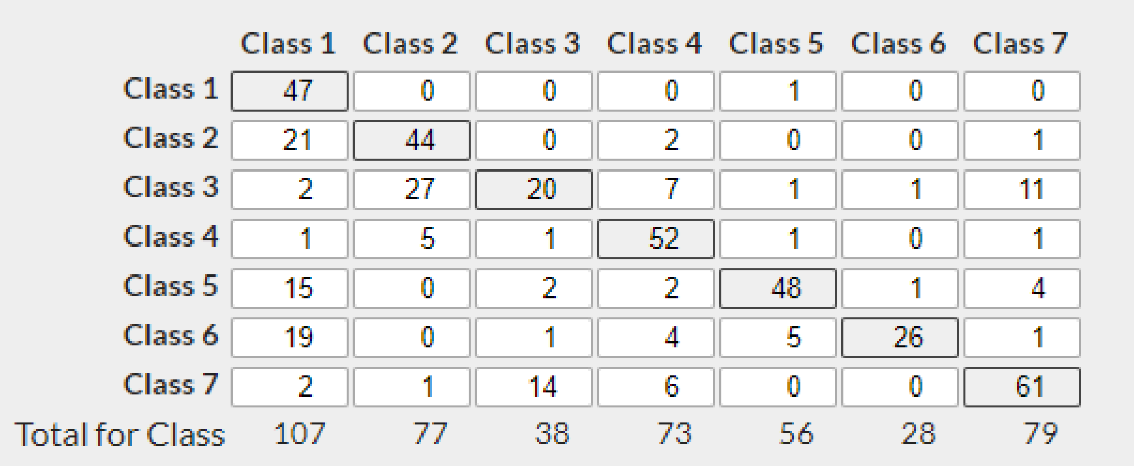
.

4.1.1 TensorFlow

The assessment of inference model on Raspberry Pi 3 and Pi 4 is compared in Tensorflow and Tensorflow Lite in order to understand the impact of hardware and software platforms to the performance of inference.

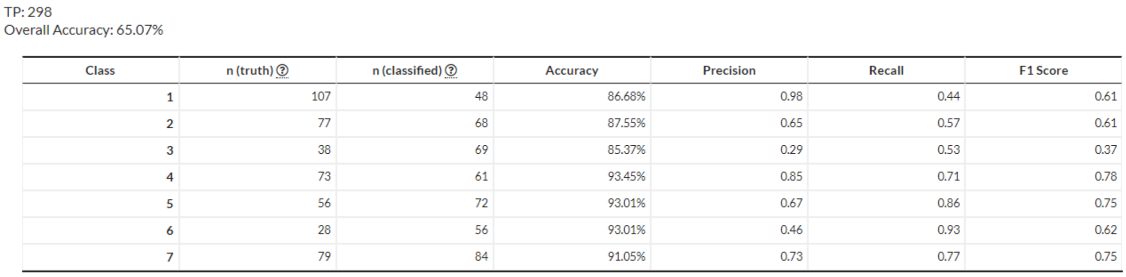
There are 7 classifications in confusion matrix with consecutively Anger, Disgust, Fear, Happy, Neutral, Sad and Surprise emotions.

The confusion matrices are calculated based on test results of model inference with test DATA.



	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
Class 1	47	0	0	0	1	0	0
Class 2	21	44	0	2	0	0	1
Class 3	2	27	20	7	1	1	11
Class 4	1	5	1	52	1	0	1
Class 5	15	0	2	2	48	1	4
Class 6	19	0	1	4	5	26	1
Class 7	2	1	14	6	0	0	61
Total for Class	107	77	38	73	56	28	79

Figure 48. *Confusion Matrix for Raspberry Pi 3 on TensorFlow*

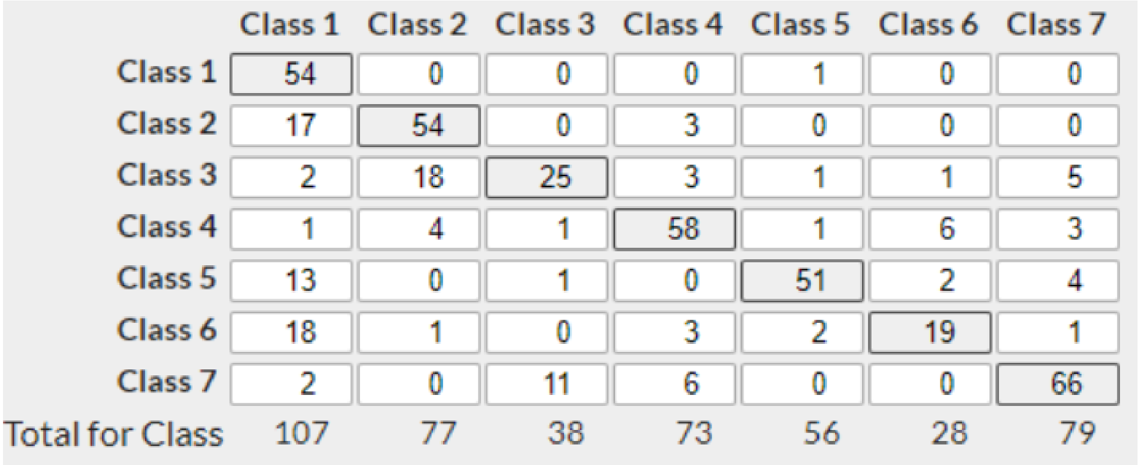


TP: 298
Overall Accuracy: 65.07%

Class	n (truth) ⓘ	n (classified) ⓘ	Accuracy	Precision	Recall	F1 Score
1	107	48	86.68%	0.98	0.44	0.61
2	77	68	87.55%	0.65	0.57	0.61
3	38	69	85.37%	0.29	0.53	0.37
4	73	61	93.45%	0.85	0.71	0.78
5	56	72	93.01%	0.67	0.86	0.75
6	28	56	93.01%	0.46	0.93	0.62
7	79	84	91.05%	0.73	0.77	0.75

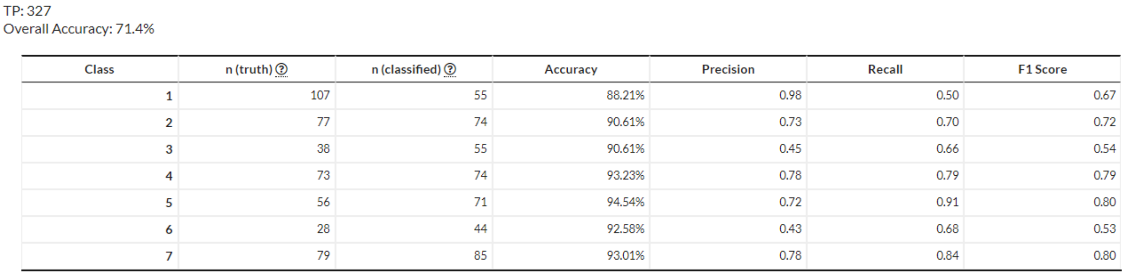
Figure 49. *Performance on Raspberry Pi 3 on TensorFlow*

As it is clear from confusion matrix and performance comparisons illustrated in figures above, the overall accuracy of model on raspberry pi 3 using Tensorflow is 65.07%. The emotion which got the best accuracy is happiness, followed by neutral, sad and others.



	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
Class 1	54	0	0	0	1	0	0
Class 2	17	54	0	3	0	0	0
Class 3	2	18	25	3	1	1	5
Class 4	1	4	1	58	1	6	3
Class 5	13	0	1	0	51	2	4
Class 6	18	1	0	3	2	19	1
Class 7	2	0	11	6	0	0	66
Total for Class	107	77	38	73	56	28	79

Figure 50. *Confusion Matrix for Raspberry PI4 on TensorFlow*



TP: 327
Overall Accuracy: 71.4%

Class	n (truth) ⓘ	n (classified) ⓘ	Accuracy	Precision	Recall	F1 Score
1	107	55	88.21%	0.98	0.50	0.67
2	77	74	90.61%	0.73	0.70	0.72
3	38	55	90.61%	0.45	0.66	0.54
4	73	74	93.23%	0.78	0.79	0.79
5	56	71	94.54%	0.72	0.91	0.80
6	28	44	92.58%	0.43	0.68	0.53
7	79	85	93.01%	0.78	0.84	0.80

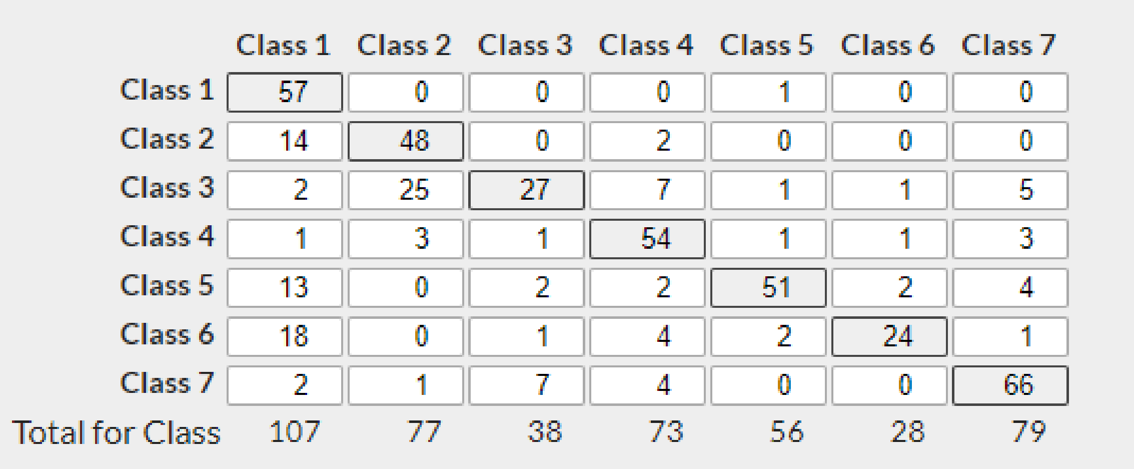
Figure 51. *Performance on Raspberry PI4 on TensorFlow*

On other hand, the accuracy of inference of model on Raspberry Pi 4 on TensorFlow is significantly higher that of Raspberry Pi 3, with 71.4 % compared to 65.07 %, respectively.

However, the best performing classifier of emotions is neutral followed by happy, sad and others.

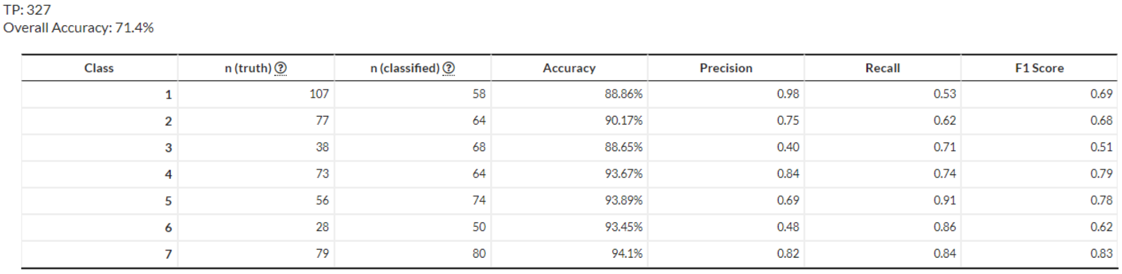
4.1.2 TensorFlow Lite

In this subsection, accuracy of the inference is tested in the lite version of Tensorflow and compared between Raspberry Pi 3 and Pi 4.



	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
Class 1	57	0	0	0	1	0	0
Class 2	14	48	0	2	0	0	0
Class 3	2	25	27	7	1	1	5
Class 4	1	3	1	54	1	1	3
Class 5	13	0	2	2	51	2	4
Class 6	18	0	1	4	2	24	1
Class 7	2	1	7	4	0	0	66
Total for Class	107	77	38	73	56	28	79

Figure 52. *Performance on Raspberry Pi 3 on TensorFlow Lite*

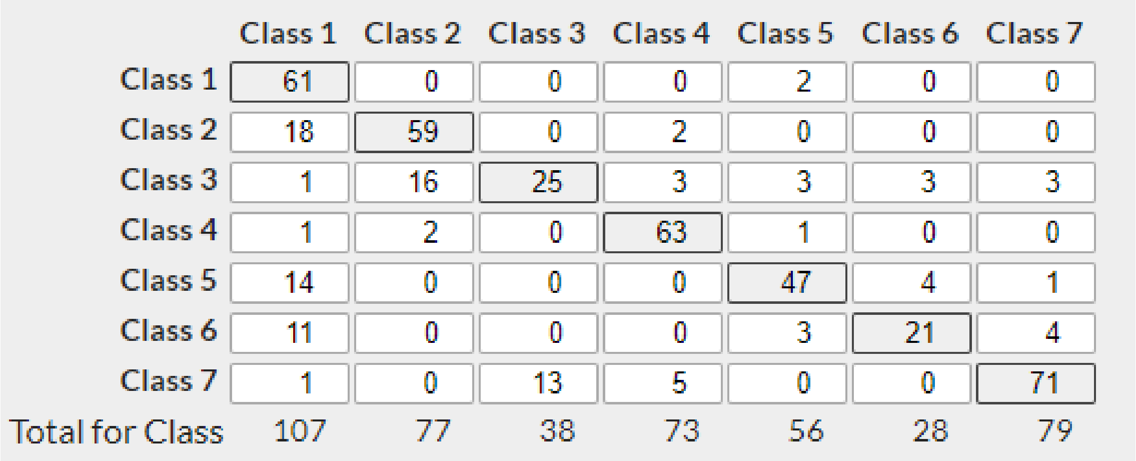


TP: 327
Overall Accuracy: 71.4%

Class	n (truth) ⓘ	n (classified) ⓘ	Accuracy	Precision	Recall	F1 Score
1	107	58	88.86%	0.98	0.53	0.69
2	77	64	90.17%	0.75	0.62	0.68
3	38	68	88.65%	0.40	0.71	0.51
4	73	64	93.67%	0.84	0.74	0.79
5	56	74	93.89%	0.69	0.91	0.78
6	28	50	93.45%	0.48	0.86	0.62
7	79	80	94.1%	0.82	0.84	0.83

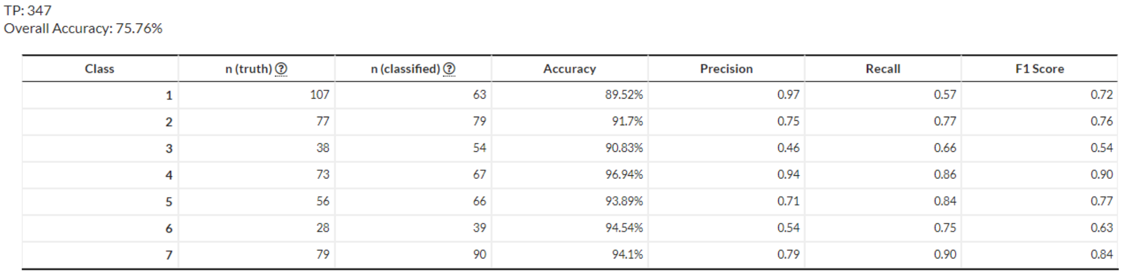
Figure 53. *Performance on Raspberry Pi 3 on TensorFlow Lite*

Based on figures above, the overall accuracy of all classifiers on Raspberry Pi 3 on Tensorflow lite, is 71.4 % which is the same as the performance of Raspberry Pi 4 but running with Tensorflow instead of lite version.



	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
Class 1	61	0	0	0	2	0	0
Class 2	18	59	0	2	0	0	0
Class 3	1	16	25	3	3	3	3
Class 4	1	2	0	63	1	0	0
Class 5	14	0	0	0	47	4	1
Class 6	11	0	0	0	3	21	4
Class 7	1	0	13	5	0	0	71
Total for Class	107	77	38	73	56	28	79

Figure 54. *Performance on Raspberry Pi 4 on TensorFlow Lite*



TP: 347
Overall Accuracy: 75.76%

Class	n (truth) ?	n (classified) ?	Accuracy	Precision	Recall	F1 Score
1	107	63	89.52%	0.97	0.57	0.72
2	77	79	91.7%	0.75	0.77	0.76
3	38	54	90.83%	0.46	0.66	0.54
4	73	67	96.94%	0.94	0.86	0.90
5	56	66	93.89%	0.71	0.84	0.77
6	28	39	94.54%	0.54	0.75	0.63
7	79	90	94.1%	0.79	0.90	0.84

Figure 55. *Performance on Raspberry Pi 4 on TensorFlow Lite*

In contrast, the accuracy of inference of model on raspberry pi 4 is slight above of 75 % which is 4 % more than that of Raspberry Pi 3 on Tensorflow Lite.

The accuracy of Raspberry Pi 4 on Tensorflow Lite dramatically outweighs the accuracy of Raspberry Pi 3 on Tensorflow.

4.2 Inference Time

Inference time is one of the crucial metrics for evaluating the performance of model. It is determined as time of getting input and making classification as an output [39].

In our experiments, Raspberry Pi Zero was highly unstable and had issues with execution. Therefore, inference time of Raspberry Pi 3 and Pi 4 on TensorFlow and TensorFlow Lite platforms are compared.

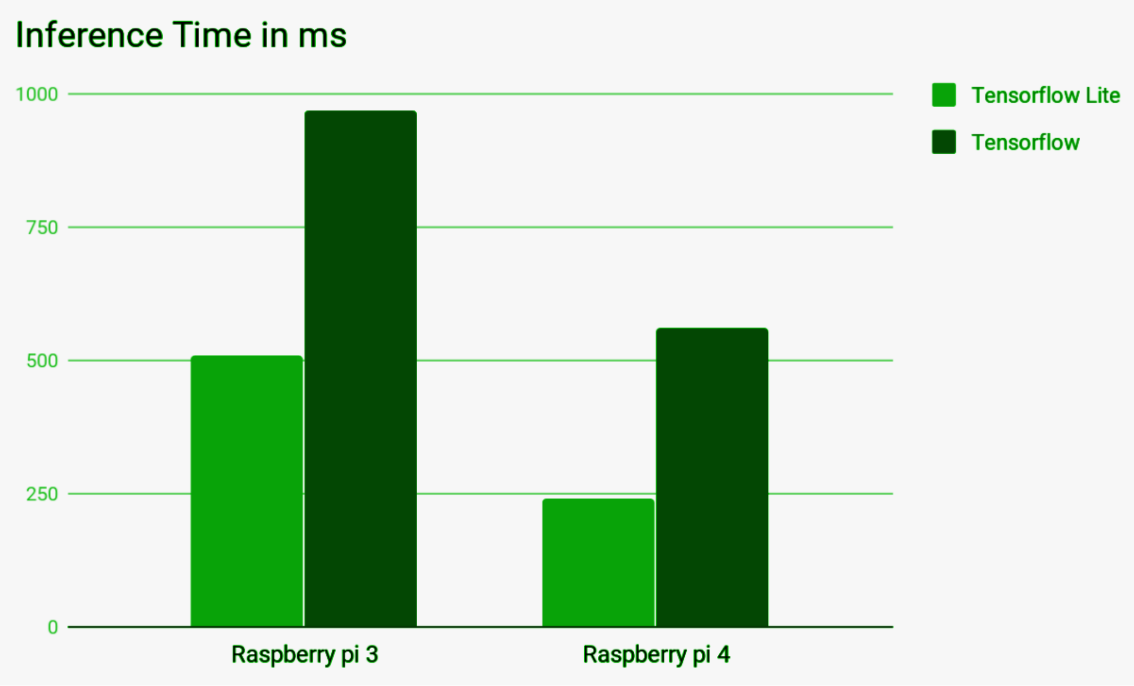


Figure 56. *Inference Time Comparison [39]*

As it is visible from diagram above, the inference time improves about twice after using Tensorflow Lite instead of Tensorflow. In addition, the inference is about twice faster on Raspberry Pi 4 on both software platforms, Tensorflow and Tensorflow Lite.

4.3 Conclusion

Based on experiments and testing of model with test data, the accuracy of Raspberry Pi 4 outperform Raspberry Pi 3 in both Tensorflow and Tensorflow Lite.

However, the change of software library from TensorFlow to TensorFlow Lite does not change the accuracy significantly, for both microprocessors the change is not more than 6 percent with 6 percent for Raspberry Pi 3 and Pi 4 percent for Raspberry Pi 4.

In contrast, the inference time doubles for both cases, when the microprocessor is changed from Raspberry Pi 3 to Pi 4 and when for each microprocessors the software library is changed from Tensorflow to Tensorflow Lite.

Finally, based on accuracy and inference time measurements of classification model, for better performance the Raspberry Pi 4 with tensorflow lite is recommended but Raspberry Pi 3 with tensorflow lite could be used to perform real time facial emotional analysis in industry too.

5. Future Work

The research helped to find answers to research questions although the work will be continued. Scope of future planned work is described below:

5.0.1 System Description

The system which supposed to solve described in thesis problem can be seen from figure below.

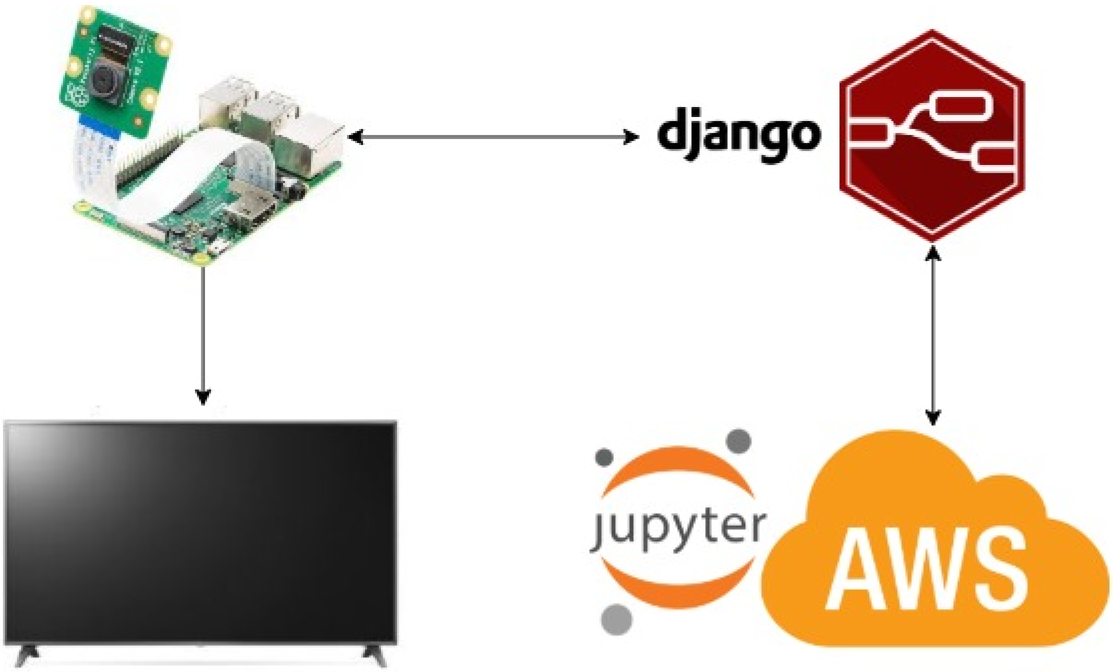


Figure 57. *The workflow of the Facial Emotions Recognition System.*

As it is observed from Figure 1-1, the Facial Emotions Recognition System consists of 4 components namely: LED/LCD monitor, Raspberry PI with the camera, Cloud Computing Service and Web App integrated with MQTT broker. LED/LCD monitor is connected with the device via USB or Internet and controlled by the device. It changes the background picture based on the device configurations. The device itself contains Raspberry PI and Web Streaming Camera. It controls the monitor by uploading the necessary image as a background theme, gathers data from Web Streaming Camera, by making initial checking like filtering the picture and sending it to the cloud computing service. Apart from that, it receives the image file from web application using MQTT or HTTP protocols, depending on performance one of them will be chosen. The images are later uploaded to the LED/LCD monitors. The Cloud Computer services which is used is AWS, Amazon Web Services. The aim of Cloud Computing is to make analysis of raw data which includes the number of people glanced on monitor, the emotions they had looking at some position and the time that they glanced by using Artificial Intelligence Software written in python OpenCV implementations of following algorithms, Eigen or Fisher Face algorithms or Local Binary Patterns Histogram. In addition, it transfers raw data to useful data which will be sent to Web Application using JSON files. The web application is going to be written using the Django framework, PostgreSQL database and will store all gathered statistics, will have a customer profile and payment system.

Artificial Intelligence software is built based on three main algorithms implemented in python using the OpenCV library. Eigen Face Algorithm, Fisher Face Algorithm, Local Binary Patterns Histogram or a mix of them. All combinations will be used and tested to find the perfect performance. The web application is written in Django framework, the database is PostgreSQL. The Cloud Computing system is chosen as Amazon Web Services. MQTT protocol is used for remote control of Raspberry Pi and Picamera via web application. The latest version of all the mentioned technologies is used.

5.0.2 Django for Web Application Development

Django is a web development framework which is python based and is useful for prototyping and building apps relatively quickly, with less code. Interestingly, this framework is named after jazz musician Django Reinhardt. Django follows Model View Template development pattern which means following:

First, HTTP request is initiated by client then URL mapper in urls.py calls related view, which views the data needed to answer to the request by appropriate model and displaying correct template. Followed by triggering correct Model which connects to database and performs necessary queries. Afterwards, template of HTML structure is generated based on Model and returned to client [40].

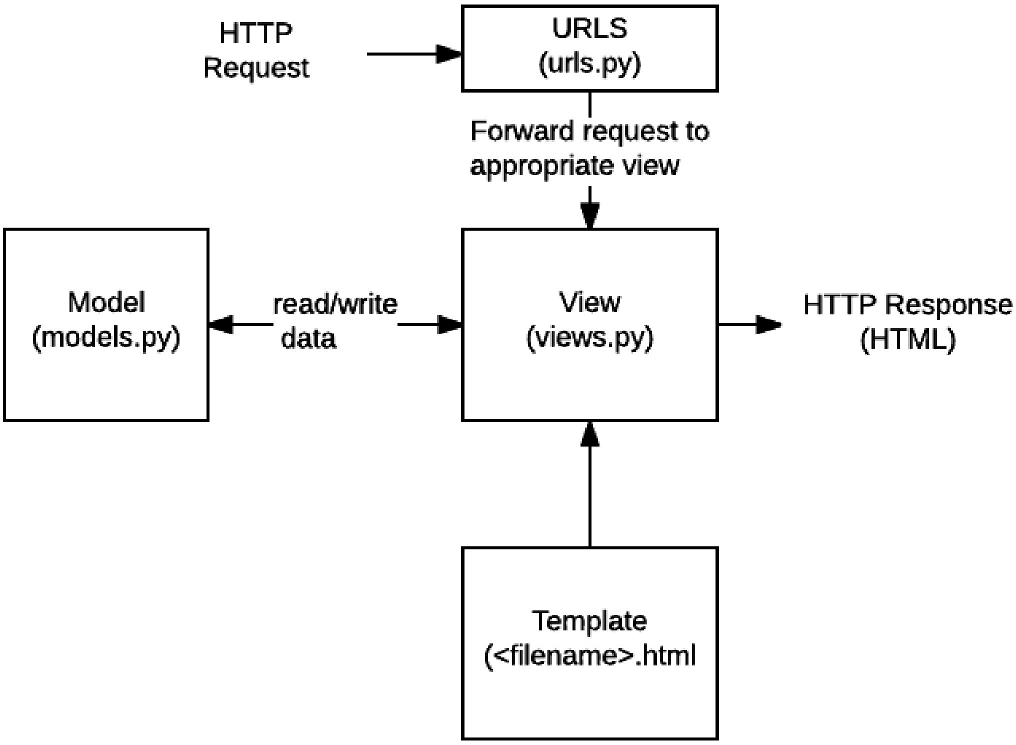


Figure 58. *Django Web Development Framework Model View Template approach [40]*

Django is very scalable and robust web development framework which is good for startups, small projects and big projects too. In addition, Django Web Development Framework has many advantages, below are some of them [41]:

- **Diverse environment**. There are many applications and functions that might be easily used by just adding and calling them. Like authorization and mail service.
- **Maturity**. Django is used for a decade and constantly evolving. Many unused services are depricated, others are improved and updated. Considering the age and maturity of framework, online support is also very rich.
- **Admin panel**.Admin panels are used to easily manage the content and structure of web applications. The Django Admin Panel is embedded in it and need no additional integration, which makes life easy. In addition, it is very flexible, so there is room to change the structure a lot.
- **ORM**. Object Relational Mapper is a library which maps web application code to the database in the form of objects.

5.0.3 Heatmap

The purpose of whole system is to gather and analyse emotions of people who glance on digital banner ads and generate heatmap indicating what emotions were triggered by looking on corners of the screen.

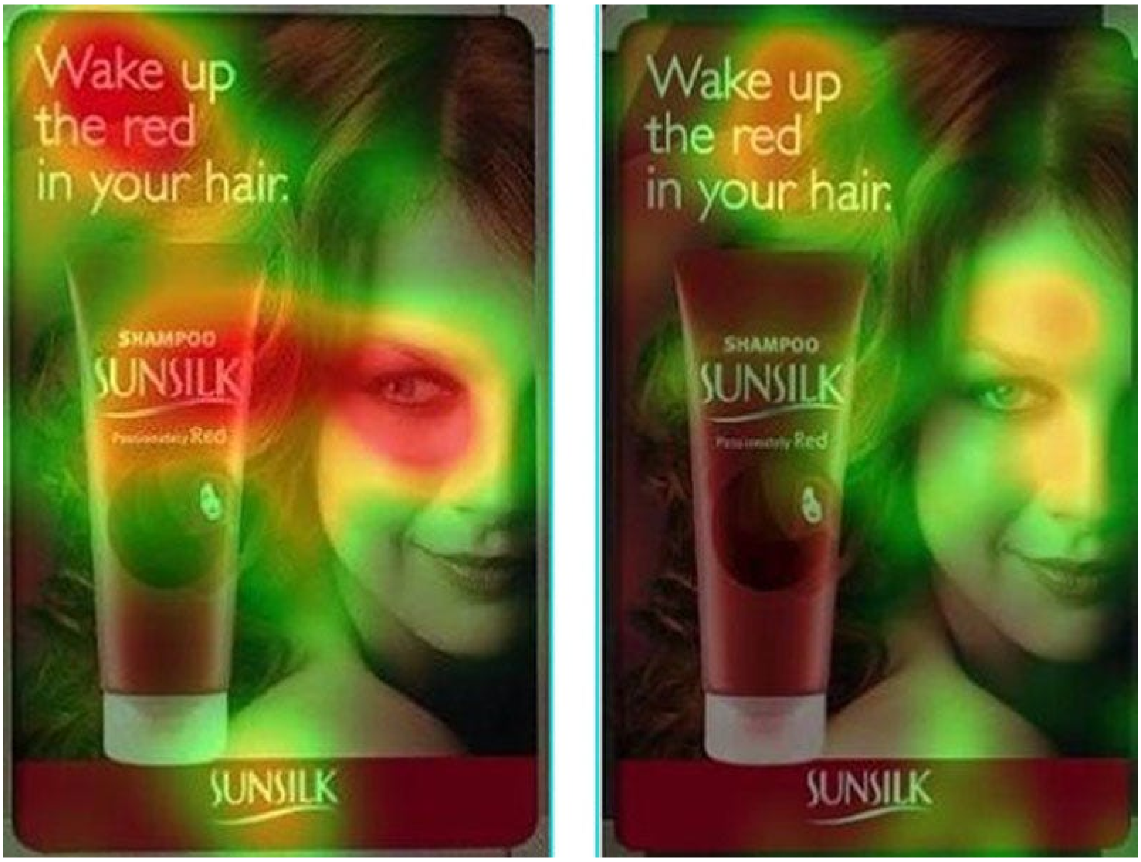


Figure 59. *Example of Heatmap which indicates the attention of people [42]*

It could be observed from figure 45 above indicating two different version of Sunsilk, that people are changing the direction of their view according to the eye movement in ads. Developed system will help business to improve their ads based on gathered analytical statistics and enhance the design of ads. The developed system will also track the movement of eyes of people who glance on banner and simultaneously it will recognise emotions, classify people by gender and age.

6. Summary

As it is clear from evaluations, Raspberry Zero is not an option at all because it failed to train the model and perform facial sentiment analysis. Oppositely, Raspberry Pi 3 with Tensorflo wLite performed well, with good precision and slightly below Raspberry Pi 4 with Tensorflow performance. The Decision was made to use Raspberry Pi 3, unless custom microprocessor will not be designed.

6.0.1 Limitations

Limitations that were found are related to broader knowledge lack in Deep Learning, Tensorflow optimization. In addition, It would be possible to accelerate computation powers of microprocessors by using Google Coral USB accelerator which accelerates inferencing of machine learning models.

6.0.2 Conclusion

Answers to research questions are found based on experiments performed, specifically the good enough microprocessor for our aims of doing facial emotional recognition is decided to be Raspberry Pi 3.

However, the work will be continued towards building web application, integrating system on chip with web application and testing it with real data.

There are strong indications that project is vital in providing analysis to digital signage marketing considering market potential and need for innovations and digitization.

Bibliography

[1] *Research methods*. [Accessed: 19-04-2020]. URL: https://guides.lib.vt.edu/researchmethods/design-method.

[2] *Digital Signage Market*. [Accessed: 14-04-2020]. URL: https://www.alliedmarketresearch.com/digital-signage-market.

[3] *Emotion Detection and Recognition Market*. [Accessed: 14-04-2020]. URL: https://www.marketsandmarkets.com/Market-Reports/emotion-detection-recognition-market-23376176.html.

[4] *Emotional Models*. [Accessed: 7-04-2020]. URL: https://www.researchgate.net/publication/319173333_Emotion_Models_A_Review.

[5] Paul Ekman. "An argument for basic emotions". In: *Cognition emotion* 6.3-4 (1992), pp. 169–200.

[6] *Plutchik Wheel of Emotions*. [Accessed: 6-04-2020]. URL: https://www.6seconds.org/2017/04/27/plutchiks-model-of-emotions/.

[7] *Ekman's six basic emotions*. [Accessed: 7-04-2020]. URL: https://news.softpedia.com/news/Is-Emotion-Tracking-the-Future-in-Tech-or-Just-Down-Right-Creepy-434806.shtml.

[8] *FER basic techniques*. [Accessed: 6-04-2020]. URL: https://www.mdpi.com/1424-8220/18/2/401/htm#cite.

[9] *Comparison of Raspberry pi 3 and zero*. [Accessed: 14-04-2020]. URL: https://www.arrow.com/en/research-and-events/articles/raspberry-pi-3-vs-raspberry-pi-zero-w.

[10] *Comparison of Raspberry pi 4 and 3*. [Accessed: 14-04-2020]. URL: https://sunrobotics.co.in/raspberry-pi-c-149/raspberry-pi-4-model-b-with-2-gb-ram?zenid=k3u6qg4991s4s12kgpmboksr44.

[11] *Machine Learning Venn Diagram*. [Accessed: 11-04-2020]. URL: https://towardsdatascience.com/notes-on-artificial-intelligence-ai-machine-learning-ml-and-deep-learning-dl-for-56e51a2071c2.

[12] *General Machine Learning*. [Accessed: 11-04-2020]. URL: https://sonix.ai/articles/difference-between-artificial-intelligence-machine-learning-and-natural-language-processing.

[13] *General AI*. [Accessed: 11-04-2020]. URL: https://www.future-of-leadership.org/2019/01/27/7-common-myths-about-ai/.

[14] Géron Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems*. OReilly Media, Inc., 2019.

[15] *Machine Learning Tree and branches*. [Accessed: 11-04-2020]. URL: https://learn.datacamp.com/career-tracks/machine-learning-scientist-with-python.

[16] Adrian Rosebrock. *Deep learning for computer vision with Python*. Adrian Rosebrock, PyImageSearch.com, 2019.

[17] *Neural Networks*. [Accessed: 11-04-2020]. URL: https://towardsdatascience.com/machine-learning-fundamentals-ii-neural-networks-f1e7b2cb3eef.

[18] *Image fundamentals*. [Accessed: 11-04-2020]. URL: https://towardsdatascience.com/understanding-images-with-skimage-python-b94d210afd23.

[19] Hongyao Deng and Xiuli Song. "Chaos-Based Image Encryption Algorithm Using Decomposition". In: *TELKOMNIKA Indonesian Journal of Electrical Engineering* 12 (July 2014). DOI: 10.11591/telkomnika.v12i1.3018.

[20] *PIL*. [Accessed: 11-04-2020]. URL: https://www.codementor.io/@innat_2k14/image-data-analysis-using-numpy-opencv-part-1-kfadbafx6.

[21] *python image library*. [Accessed: 11-04-2020]. URL: https://pillow.readthedocs.io/en/stable/.

[22] *VENV*. [Accessed: 11-04-2020]. URL: https://medium.com/@peterchang_82818/python3-beginner-must-know-venv-fdbbd0421405.

[23] *Tensorflow Lite*. [Accessed: 11-04-2020]. URL: https://www.tensorflow.org/lite/convert.

[24] *Viola Jones*. [Accessed: 11-04-2020]. URL: https://medium.com/0xcode/the-viola-jones-face-detection-algorithm-3eb09055cfc2.

[25] *Viola Jones*. [Accessed: 11-04-2020]. URL: http://www.ipol.im/pub/art/2014/104/?utm_source=doi.

[26] *Viola Jones*. [Accessed: 11-04-2020]. URL: https://www.udemy.com/course/python-for-computer-vision-with-opencv-and-deep-learning/.

[27] Active Shape Models. *Active Shape Model*. 2004.

[28] *Dawin on Emotions*. [Accessed: 11-04-2020]. URL: https://medium.com/analytics-vidhya/facial-emotion-classification-using-deep-learning-d08dd02a2d38.

[29] *Raspberry Pi OS*. [Accessed: 11-07-2020]. URL: https://www.raspberrypi.org/downloads/raspberry-pi-os/.

[30] *Burn OS with Etcher*. [Accessed: 11-07-2020]. URL: https://www.balena.io/etcher/.

[31] *Connect SD*. [Accessed: 11-07-2020]. URL: https://projects.raspberrypi.org/en/projects/raspberry-pi-getting-started/3.

[32] *VNC*. [Accessed: 11-07-2020]. URL: https://www.realvnc.com/en/connect/download/viewer/.

[33] *Pi Camera*. [Accessed: 11-07-2020]. URL: https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/2.

[34] *USB Camera*. [Accessed: 11-07-2020]. URL: https://www.raspberrypi.org/documentation/usage/webcams/.

[35] *Confusion Matrix*. [Accessed: 11-04-2020]. URL: https://www.geeksforgeeks.org/confusion-matrix-machine-learning/.

[36] *Confusion Matrix*. [Accessed: 11-04-2020]. URL: https://machinelearningmastery.com/confusion-matrix-machine-learning/.

[37] Sebastian Bittrich, Michael Schroeder, and Dirk Labudde. "StructureDistiller: Structural relevance scoring identifies the most informative entries of a contact map". In: *Scientific Reports* 9.1 (Dec. 2019). DOI: 10.1038/s41598-019-55047-4. URL: https://doi.org/10.1038/s41598-019-55047-4.

[38] *Evaluation of Models*. [Accessed: 14-04-2020]. URL: https://www.jeremyjordan.me/evaluating-a-machine-learning-model/.

[39] *Inference Time*. [Accessed: 11-04-2020]. URL: https://deci.ai/the-correct-way-to-measure-inference-time-of-deep-neural-networks/.

[40] *Django*. [Accessed: 3-04-2020]. URL: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction.

[41] *Django Advantages*. [Accessed: 3-04-2020]. URL: https://steelkiwi.com/blog/why-django-best-web-framework-your-project/.

[42] *Example of Heatmap*. [Accessed: 14-04-2020]. URL: https://pgpneuromarketing.wordpress.com/sabia-que/eye-tracking-heatmaps-reveal-where-people-really-look/.