PROJECT REPORT ON

IoT Based Health Monitoring in smart home system

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
INFORMATION AND COMMUNICATION TECHNOLOGY

Submitted by

118014048-Kuppa Venkata Naga Abhiram-ICT

Under the Guidance of

Prof. Dr D.Manivannan
SAP/CSE

School of Computing
SASTRA DEEMED TO BE UNIVERSITY
(A University established under section 3 of the UGC Act,1956)
Tirumalaisamudiram
Thanjavur-613 401

April (2018)

SHANMUGHA
ARTS, SCIENCE, TECHNOLOGY & RESEARCH ACADEMY
(SASTRA DEEMED TO BE UNIVERSITY)
(A University Established under section 3 of the UGC Act, 1956)
TIRUMALAISAMUDRAM, THANJAVUR – 613401

BONAFIDE CERTIFICATE

Certified that this project work entitled "**IoT BASED HEALTH MONITORING IN SMART HOME SYSTEM"** submitted to the Shanmugha Arts, Science, Technology & Research Academy (SASTRA Deemed to be University), Tirumalaisamudram -613401 by 118014048 - Kuppa Venkata Naga Abhiram – ICT, in partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in **INFORMATION AND COMMUNICATION TECHNOLOGY**. This work is an original and independent work carried out under my guidance, during the period December 2017 - April 2018.

Dr D. MANIVANNAN	**ASSOCIATE DEAN**
	DR. K S RAVICHANDRAN
SCHOOL OF COMPUTING	**SCHOOL OF COMPUTING**

Submitted for Project Viva Voice held on____________________

Examiner –I **Examiner-II**

ACKNOWLEDGEMENT

We would forever remain grateful to honourable **Prof. R. Sethuraman, Vice Chancellor** for his encouragement in our academic life at SASTRA University.

We wish to express my profound gratitude **to Dr. S. Vaidhyasubramaniam, Dean - Planning & Development and Dr S. Swaminathan ,Dean Sponsored research** for his overwhelming support provided during our course span in SASTRA University

We are extremely thankful to **Dr. K. S. Ravichandran,** Associate Dean, Department of Information and Communication Technology for providing us the opportunity to do this project and for all the academic help done for the past three years.

We sincerely express our gratitude to our mentor **Dr D Manivannan**, Senior Associate Professor ,School of Computing for his assistance and guidance for the successful implementation of project in a systematic and professional manner.

We also thank all the Teaching and Non-teaching staff, and those who have directly or indirectly helped us by extending their moral support and encouragement for completion of this project.

ABSTRACT

KEY WORDS: Internet of Thing, Raspberry Pi, Emotion recognition, health monitoring.

The number of patients seeking treatment is always on the rise, while hospitals are providing treatment with specialized medical facilities and nursing staffs. With the advent of Internet of things and its applications patient can choose the place of treatment based on the convenience. By making use of advances in IoT, hospitals can provide patients with monitoring system for supervision without the need of separate medical expert for each patient . This work recommends a health monitoring system which has a vigilance on the emotion of the patient to detect pain through facial expressions while continuously checking for vitals in a cost effective manner . Although images are used for ensuring safety, a very little use of it has been found in the field of health monitoring.With images helping out medical experts to know the feelings of the patient providing them a more concentrated care in cost effective manner , IoT enabled embedded hardware based health monitoring system with camera interface is proposed and implemented in this work .

LIST OF DIAGRAMS

Sl No.	Contents	Page No.
1.1	Russell’s Circumplex wheel	4
4.1	Work Flow Model	9
4.2	Use Case Diagram	10
4.3	Entity Relationship Diagram	10
5.1	Design of entire system	12
7.1-7.7	Emotions Unit Test	26,27
7.8	System testing Result	28

TABLE OF CONTENTS

Page No

ACKNOWLEDGEMENTS ……….…………………………………...………………. (iii)

ABSTRACT……………………………………………… ……………..……………(iv)

LIST OF DIAGRAMS………………………… ……………….. ……………………(v)

CHAPTER 1 INTRODUCTION …………………..………......................................…(1)

CHAPTER 2 PROJECT PLAN …………………………………...……………….(6)

CHAPTER 3 REQUIREMENTS SPECIFICATION

3.1 Functional requirements …………………………………………………………(7)

3.1.1 Use Case Diagram…………………………………………………...(7)

3.1.2 Flow Chart…………………………………………………………...(7)

3.2 Non-functional requirements……………………..……………………………… (7)

3.2.1 Resources requirements (hardware & software)…………..……..………..(7)

3.2.2 Operational requirements…………………………………………….(8)

3.2.3 Performance Requirements…………………………………………..(8)

CHAPTER 4 SYSTEM ANALYSIS

4.1 Data Flow Diagram.. ………..…………………………………...………………(9)

4.2 Use case diagram………....……………………………………...…………… (10)

4.3 Entity Relationship Diagram..…..……………………………...…………… (10)

CHAPTER 5 DESIGN

5.1 Sensor Module….. ……………..……………………………….………………… (11)

5.2 Emotion Recognition Module…..…………………………….…………………. (12)

5.3 Web Service Module………..……………………………….…………………. (13)

CHAPTER 6 CODING

6.1 Algorithm/Pseudo Code ……………………….…………………………..….(14)

6.2 Complete Code…………………………………………………………. (15)

CHAPTER 7 TESTING

7.1 Unit Testing……………………………….………………..…….… (26)

7.2 Integration Testing…………………………..……………………..…(28)

7.3 System testing……………………………………………………… (28)

CHAPTER 8 IMPLEMENTATION

8.1 Running The System…………………………………………………...(32)

8.2 Problems Faced ………………………………….………………… (32)

CHAPTER 9 CONCLUSION AND FUTURE PLANS ………………………….. (33)

REFERENCES…………………………………….……………………… **..**(34)

CHAPTER 1

INTRODUCTION

Internet is one of the widely used and one of the greatest inventions in the past century . It provides variety of information and communication facilities to people all over the world .

With internet wide range of technologies have come along and Internet of Things(IoT) is one of them . Internet of Things means a network of physically connected devices which are enabled to connect and exchange data . Though each device may act uniquely they are able to inter operate within existing internet infrastructure .

Though IoT has been used in many number of fields to make life easy , its use in the field of health monitoring is very limited . Health, has become a major concern in the modern day . So by making use of advances in IoT , hospitals can provide patients with monitoring system for supervision without the need for a separate medical expert for each patient .

The introduction throws a light on the related works that have been carried out until now and how the system uses them to create a health monitoring system along with emotion recognition system to detect pain .

There may be people who may not afford a full-time service of a caretaker or a nurse . Which can lay the ground for technology to come to the rescue and provide a full-time surveillance of the patient and alert the caretaker or relatives whenever necessary . This way an elderly person who may live alone can be taken care or treated at home without spending much time at the hospital.

In the light of the above events a system must be present such that it identifies the patients vitals and stream the real time data to the caretaker or nurse and alert them when needed . Some systems already exists which where developed to detect the vitals of a patient using an low power microcontroller system such as arduino which combine a group of sensors to provide real time vitals monitoring of the patient.

Not only sensors but facial expressions of patients can also tell about the condition of the patient health . Here images of patients face are used to detect any kind of pain in their body . Although the use of images for ensuring safety is used widely , a very little use of it has been found in the field of health monitoring . With images helping out medical experts to know the feelings of the patient providing them a more concentrated care , which can be cheap and easy with the help of IoT technologies.

Emotions are the fundamental responses to external events that happen around us . Emotions include facial expressions, hand gestures and tone of voice which tells how a person feels and reacts, it is often intertwined with mood, temperament, personality, disposition, and motivation . Since the last decade many people have been experimenting on new ways and different sophisticated algorithms with image processing techniques to identify what an image or a video of a particular person can say about his or her feelings using machine learning techniques . Understanding these emotion presents a great deal of information on the persons present health status.

Researches have been trying to investigate techniques so that a machine can detect the way a person feels by these small gestures . While the take here is to map points onto a recognized face and based on calculations , provide output the emotion of a person with the input being the face of the patient.

According to Ekman & Friesen *facial behaviors are universally associated with particular emotions* . All Humans develop similar kind of facial muscular features irrespective of their place of birth,color or race . Based on the above statements they classified emotions into six basic types which are (anger , fear , surprise , disgust , happy , sad) . Through Machine learning(ML) the machine can be taught to detect emotions, from which we may determine whether a person is experiencing pain.

The wheel below known as Russell's circumplex model which states that each emotion can be understood as a linear combination of these two dimensions , or as varying degrees of both valence and arousal . The Fig.1.1 shows the six basic emotions on the outer circle from which , we can identify the feelings that are present inside of it.

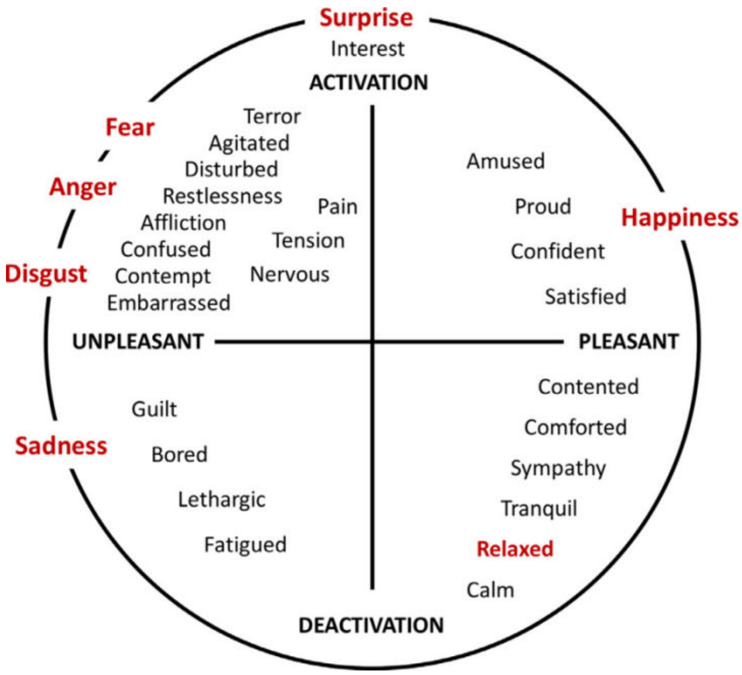

Fig 1.1.-Russell's Circumplex wheel describing basic emotions

Experimental results have shown that finding happy and sad are way more easier that the rest , while fear and disgust being way more difficult . The implementation of different algorithms are show in other papers , The results show SVM(Support Vector Machine) performing far more accurate than its peers . Leaving with modules that use SVM to detect emotions.

Specially when the systems are installed at homes it may raise the issue of privacy but that camera can be made online or offline , by turning them on or off and that does depends on the wish of the patient getting treated.

Since the last decade sensor have become a major part of life as they have been used every where lately in phones , cars , smart homes etc. . These sensors come in handy where monitoring has to be done in real time . Which means sensors must be running quickly , smoothly while having a long life and when these sensors stop functioning properly the particular device is doomed by giving negative results causing a failure of the device.

With technology the medical field has seen a quite share of sensors for real time monitoring patients and their vitals . Sensors along with a controller(microprocessors or micro controllers) can perform special tasks and produce alerts when necessary . Every sensor has its own task which it performs in-turn to produce data.

A system in IoT generally has a cloud which stores and displays the data . Services as the above are provided by companies like IBM , Amazon etc giving developers a easy way to store and display data . Amazon web service IoT is one such service provided by amazon for establishing connections between the system and the cloud.

CHAPTER 2

PROJECT PLAN

2.1. System Plan:

Literature survey	-	10 DAYS
System design	-	8 DAYS
Coding and execution	-	35 DAYS
Testing	-	9 DAYS
Project report	-	10 DAYS

CHAPTER 3

REQUIREMENTS SPECIFICATION

3.1 .Functional requirements:

3.1.1 Use Case Diagram

Purpose of use case diagram is

- To know the actors of the system.
- To understand the behaviour of the whole system.
- To identify the test cases.
  - To distribute workload to different modules.
  - To create test cases for each module.

3.1.2 Flow Charts

- To describe the data creation.
- To describe the data flow from creation till the end user.
- To identify single point errors.
- Identify test cases.

3.2 .Non Functional Requirements:

3.2.1 Resource requirements :

1. Raspberry Pi 3
2. Arduino Uno
3. Pi Camera

4.Sensors of choice(ultrasonic,temperature,pulse sensors)

5.Memory card for pi with raspian installed.

6.A working wifi connected with raspberry pi.

3.2.2 Operational requirements:

1.A memory card with same installations as above for back up as memory cards can get corrupted.

3.2.3 Performance requirements :

Dependencies to be installed on raspberry pi are,the modules with versions have to be the same as given below rest can be the latest versions.

1. python 2.7 or 3.4/higher
2. Keras - 2.0.5
3. Matplotlib
4. Numpy
5. Tensorflow - 1.1.0
6. Opencv - 3.3.0
7. h5py
8. Pandas - 0.22.0
9. Pip

CHAPTER 4

SYSTEM ANALYSIS

4.1 Data Flow Diagram

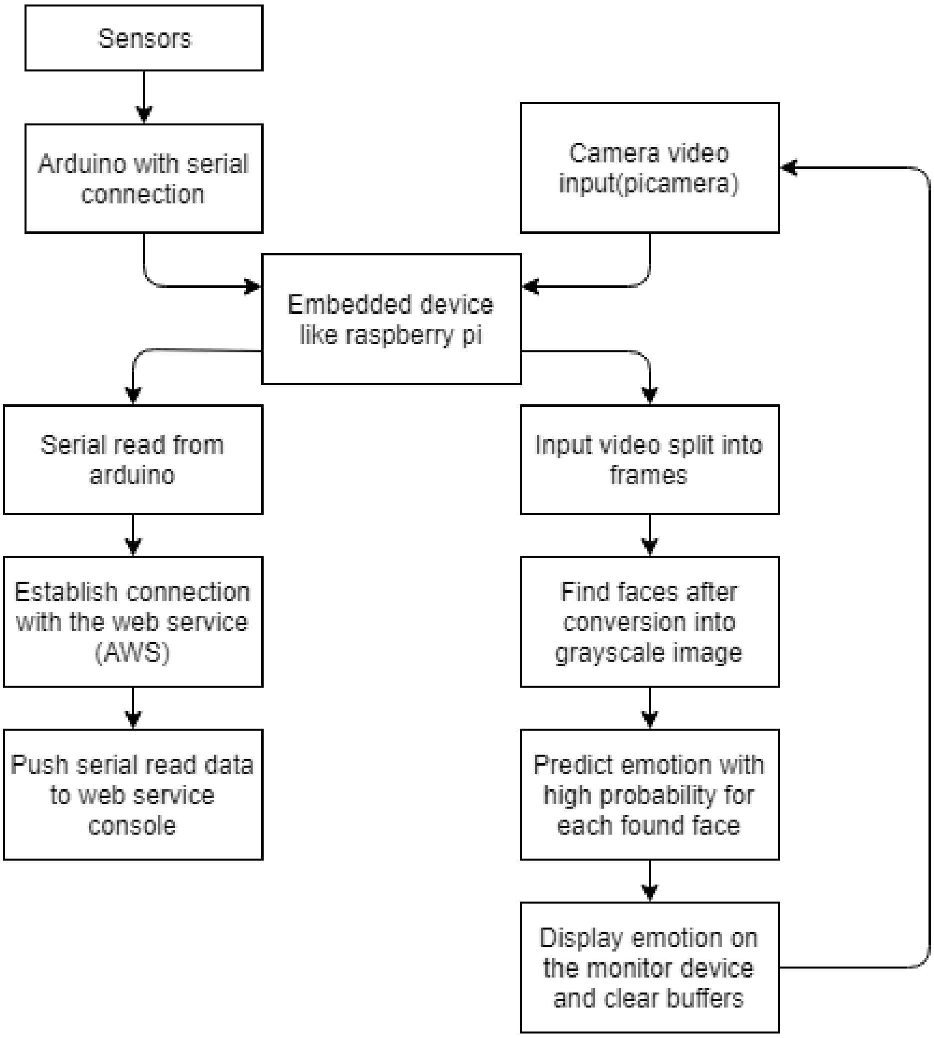

Fig.4.1.Workflow model of the system.

The Fig 2. represents Work flow of the entire system from data creation to data display on the console.

4.2 Use Case Diagram:

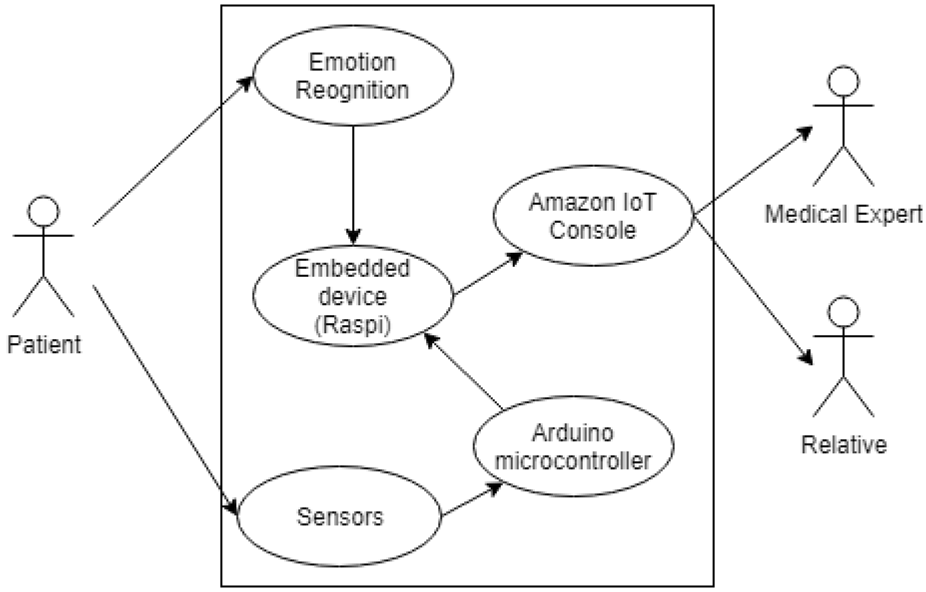

Fig.4.2.Use Case Diagram System.

4.3 Entity Relationship Diagram

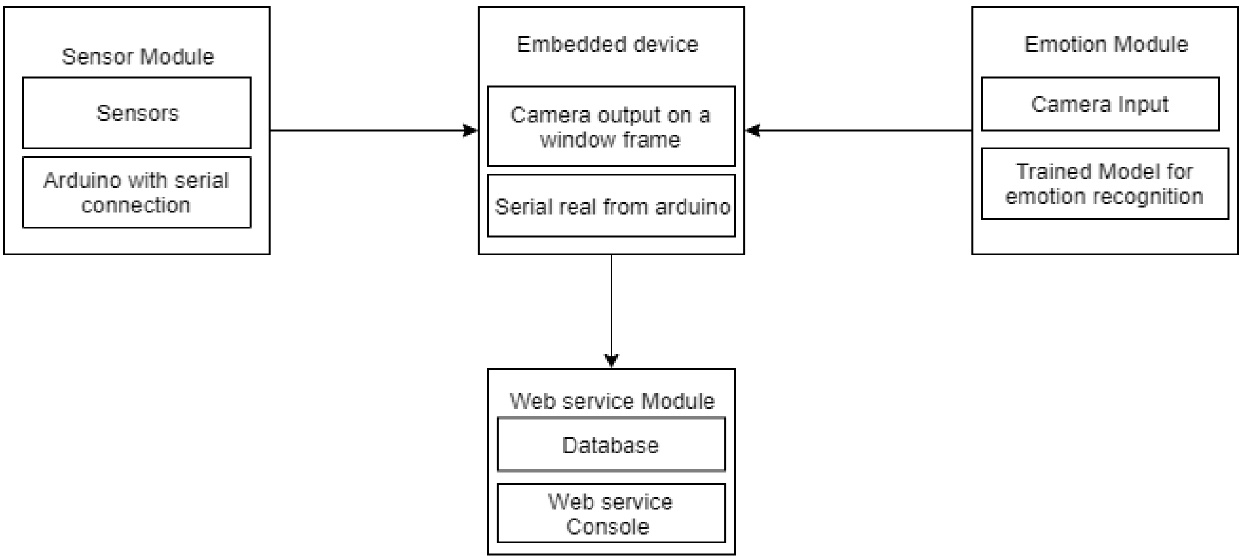

Fig.4.3.Overview of system block diagram.

CHAPTER 5

DESIGN

The Design of the recommended system was divided into 3 modules namely

1. Sensor Module   2 . Emotion recognition Module   3. Web Service Module

1)Sensor module as the name suggests uses sensors to monitor the vitals of the patient in real time . The system uses arduino to act as a central unit to the sensors . Arduino is a single board microcontroller which can be used to build digital devices or units which can sense and control objects in the physical and digital world . Purpose of choosing arduino here is because of its simplicity , low cost and is very easy to code . Sensors are devices which sense the surroundings to gain information of surrounding . As of writing there are diseases which many people are suffering from and every diseases has its way to recover from , which means all treatments need not have specific sensors.

The selection of sensors is purely based on the type of treatment which depends on diseases the patient has . The system provides details on some of the basic sensors that where used during prototyping the system.

For blood or saline monitoring an ultra sonic sensor mounted onto the top of the bottle with the face of the sensor towards the liquid inside the bottle , constantly giving the height of the liquid left was used . Whenever the height went down to value lower than the threshold the central unit arduino throws an alert . The arduino must be acquainted with the threshold for all the sensors to create the alert when needed . A temperature sensor was used to identify the body temperature of the patient . A live and running heart beat must always be there for life to be present , a pulse sensor was used to provides live heartbeats and BPM's of the patient at all times . An IR sensor was used to check for any clumps during the transfusion.

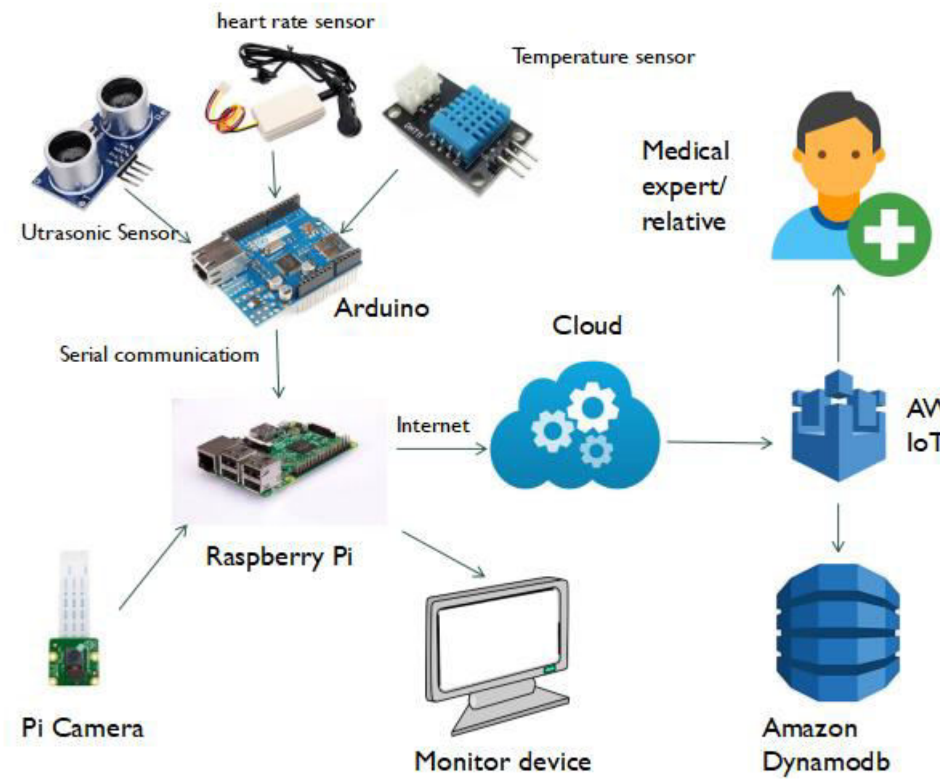

Fig 5.1. Design of the entire system

The addition of additional sensors can be done just by import the libraries into and providing the threshold for the sensor.

.

2) In related works above many theories on recognition of emotion and the six basic emotions have been discussed , so as to detect these emotions we used the emotion recognition module which has a picamera to take video input which is the patient's face . Picamera is a 8-megapixel sensor and can capture video at about 90fps with a

1280*720 resolution . The raspberry pi is a low processing power system on chip computer . Providing user with a desktop like interface . The raspberry pi is used to display emotion on the monitor device connected to it . For that the pi must be on the same network as of the monitoring device . More on the working of the module will be discussed below .

3) Web service module acts as a display unit for the sensors data . Making use of the Amazon web services IoT interface the sensor data is relayed and displayed in the AWS console . A person with the authorization can view the data at all times which is relayed to the cloud .

CHAPTER 6

CODING

6.1 Algorithm / Pseudocodes

6.1.1 Emotion Detection:

1.Initialize necessary parameters

2.while true

   1.Take face input image

   2.convert image to gray scale

   3.find faces and resize them

   4.for each found face

      1.detect emotion through model

   5.End loop

   6.display emotion in window frame and terminal

   7.Empty buffer for next images

3.End loop

4.End

6.1.2 Web Service display the data from arduino

1.Initialize dependencies

2.Specify port to serial read from

3.Initialize client id

4.Amazon Iot provides an endpoint, to be specified

5.Download certificates and provide their path

6.connect with MQTT

7.Publish data to created thing id

8.while(till interrupted)

    1.Serial read from port

    2.Publish the read data to AWS

    3.Display on Console

9.End loop

10.End

6.2 Complete Code

6.2.1 Code to detect emotions(Emotion Recognition):

Name: emotion.py

```
#import necessary packages specified above

import cv2

import numpy as np

from keras.models import load_model

from statistics import mode

import matplotlib as mpl

mpl.use('PDF')

from utils.datasets import get_labels

from utils.inference import detect_faces

from utils.inference import draw_text
```

```
from utils.inference import draw_bounding_box

from imutils.video import VideoStream

from utils.inference import apply_offsets

from utils.inference import load_detection_model

from utils.preprocessor import preprocess_input

import imutils

import time

from picamera.array import PiRGBArray

from picamera import PiCamera

#end of import


#provide path for the trained model

emotion_model_path = './models/emotion_model.hdf5'


#getting the labes of the emotions from utils folder

emotion_labels = get_labels('fer2013')


#set up frame window and emotion offset size

frame_window = 10

emotion_offsets = (20, 40)


#provide path for xml file that detects faces in each of the frame

face_cascade =

cv2.CascadeClassifier('./models/haarcascade_frontalface_default.xml')
```

```
#load the model whose path is given above (.hdf5 file)

emotion_classifier = load_model(emotion_model_path)


#set up target size of the input shape

emotion_target_size = emotion_classifier.input_shape[1:3]


#initialize camera and variables for future use

emotion_window = []

cap = None

camera = PiCamera()

camera.resolution = (640, 480)

camera.framerate = 25

rawCapture = PiRGBArray(camera, size=(640, 480))

#end of initialization


#sleep time to load camera

time.sleep(0.1)


#loop to detect emotions continuously for each frame

for cap in camera.capture_continuous(rawCapture, format="bgr",

use_video_port=True):


        #capture single frame

        bgr = cap.array
```

```
    #conversion of found image to gray scale image
    gray_image = cv2.cvtColor(bgr, cv2.COLOR_BGR2GRAY)
    rgb = cv2.cvtColor(bgr, cv2.COLOR_BGR2RGB)

        #use the gray scaled image to find faces in the image using the xml file loaded
above
    faces = face_cascade.detectMultiScale(gray_image, scaleFactor=1.1,
minNeighbors=5,
                            minSize=(30, 30), flags=cv2.CASCADE_SCALE_IMAGE)
        #faces has all the faces present in the frame captured

    k=0
        #for all found faces perform emotion recognition
    for face in faces:

                #resizeing the face to target size
        x1, x2, y1, y2 = apply_offsets(face, emotion_offsets)
        gray_face = gray_image[y1:y2, x1:x2]
        try:
            gray_face = cv2.resize(gray_face, (emotion_target_size))
        except:
            continue
        gray_face = preprocess_input(gray_face, True)
        gray_face = np.expand_dims(gray_face, 0)
        gray_face = np.expand_dims(gray_face, -1)
```

```
            #predict emotions probability of the resized face found
    emotion_prediction = emotion_classifier.predict(gray_face)

            #the above statement gives probability of each emotion and we choose
emotion which having the highest among them
    emotion_probability = np.max(emotion_prediction)

            #getting label for the predicted emotion
    emotion_label_arg = np.argmax(emotion_prediction)
    emotion_text = emotion_labels[emotion_label_arg]
    emotion_window.append(emotion_text)

            #labeling each face with numbers
    k=k+1



    if len(emotion_window) > frame_window:
      emotion_window.pop(0)
    try:
      emotion_mode = mode(emotion_window)
    except:
      continue

            #print predicted emotion to the screen
```

```
print (emotion_text,k)

        #using different colors for various emotions to draw a box around the face for displying on the screen
        if emotion_text == 'angry':
    color = emotion_probability * np.asarray((255, 0, 0))
elif emotion_text == 'sad':
    color = emotion_probability * np.asarray((0, 0, 255))
elif emotion_text == 'happy':
    color = emotion_probability * np.asarray((255, 255, 0))
elif emotion_text == 'surprise':
    color = emotion_probability * np.asarray((0, 255, 255))
else:
    color = emotion_probability * np.asarray((0, 255, 0))

color = color.astype(int)
color = color.tolist()

        #drawing the box around the face for each face found
draw_bounding_box(face, rgb, color)

        #display emotion text above the box
draw_text(face, rgb, emotion_mode,
          color, 0, -45, 1, 1)
```

```
        #display the window with all the above features
    bgr = cv2.cvtColor(rgb, cv2.COLOR_RGB2BGR)
    cv2.imshow('window_frame', bgr)

        #To stop reading emotions press q on the window frame
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

        #empty the buffer for the next frame
    rawCapture.truncate(0)

#close all windows
cv2.destroyAllWindows()
```

6.2.2 Code to read sensor data with arduino:

Upload the following on the arduino by connecting arduino to a laptop when connected to pi it runs the code automatically

```
#include <dht.h>
const int pingPin = 7;    //ultra sonic sensor tigger pin
const int echoPin = 6;    //echo pin
#define Heart 2         //Heart beat pin
dht DHT;
#define DHT11_PIN 3
boolean beat = false;
```

```
//Initialize the above variables
//Runned only once
void setup()
{
analogReference(INTERNAL);
pinMode(Heart, INPUT);
Serial.begin(9600);
}

void loop()
 {
//Saline monitoring system code
 long duration=0,h=0;
pinMode(pingPin, OUTPUT);
digitalWrite(pingPin, LOW);
delayMicroseconds(2);
digitalWrite(pingPin, HIGH);
delayMicroseconds(10);
digitalWrite(pingPin, LOW);
pinMode(echoPin, INPUT);

duration = pulseIn(echoPin, HIGH);
h=duration/58;
h=25-h;
```

```
//Whenever height of saline becomes lower than the below value the remaining
height will be displayed
if(h<10)
{
  digitalWrite(13,HIGH);
  Serial.print("   the height of the saline remaining is    ");
  Serial.print(h);
  Serial.print("cm    ");
}

//Checks whether the heart is beating or not
if(digitalRead(Heart)>0){
    if(!beat){
      beat=true;
      Serial.print("    present heart beat");
    }
}

//To check the temperature and humidity of the patient and surroundings respectively
    int chk = DHT.read11(DHT11_PIN);
  Serial.print("Temperature = ");
  Serial.print(DHT.temperature);
  Serial.print("     Humidity = ");
  Serial.println(DHT.humidity);
```

```
//display data with 1 sec delay
delay(1000);
}
```

6.2.3 Code to stream data over to aws:

Name :Publish.py

```
#Import Needed dependencies
from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient
from time import sleep
from datetime import date, datetime
import serial

#specify raspberry pi with port and baud rate it must read data from
ser = serial.Serial('/dev/ttyACM0',9600)
ser.baudrate=9600
# AWS IoT certificate based connection
myMQTTClient = AWSIoTMQTTClient("123afhlss456")
myMQTTClient.configureEndpoint("a27pb8hmkx5vf5.iot.us-east-
1.amazonaws.com", 8883)
myMQTTClient.configureCredentials("/home/pi/cert/CA.pem",
"/home/pi/cert/d96a287d20-private.pem.key", "/home/pi/cert/d96a287d20-
certificate.pem.crt")
myMQTTClient.configureOfflinePublishQueueing(-1)  # Infinite offline Publish
queueing
```

```
myMQTTClient.configureDrainingFrequency(2)  # Draining: 2 Hz
myMQTTClient.configureConnectDisconnectTimeout(10)  # 10 sec
myMQTTClient.configureMQTTOperationTimeout(5)  # 5 sec

#connect and publish
myMQTTClient.connect()
myMQTTClient.publish("thing01/info", "connected", 0)

#loop and publish sensor reading
while 1:
    #read date and time
      now = datetime.utcnow()
       now_str = now.strftime('%Y-%m-%dT%H:%M:%SZ') #e.g. 2016-04-
18T06:12:25.877Z
       #read from serialport of arduino
    read_serial=ser.readline()
      #create a string to display on the aws console
    payload = '{  "timestamp": "' + now_str + '", ' + str(read_serial) + '}'
      #print data onto the terminal
      print payload
      #publish by giving the thing name to subscribe to
    myMQTTClient.publish("thing01/data", payload, 0)
       sleep(2)               #repeat above for every 2 seconds
```

CHAPTER 7

TESTING

Each modules are tested separately and together and below are the results

7.1 Unit Testing

Unit Testing is a level of software testing where individual units/ components of a software are tested . The purpose is to validate that each unit of the software performs as designed . A unit is the smallest testable part of any software . It usually has one or a few inputs and usually a single output .

Testing of each emotion predictability(Unit testing).

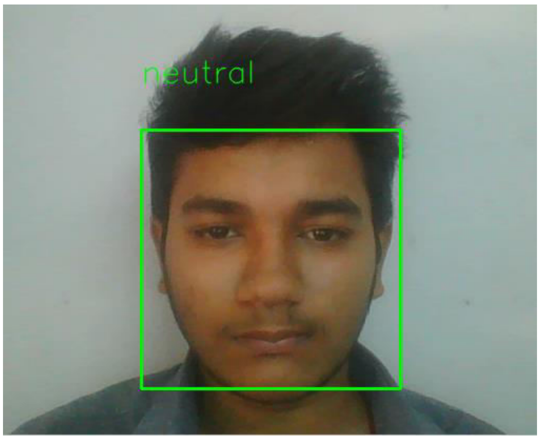

Fig .7.1.Neutral

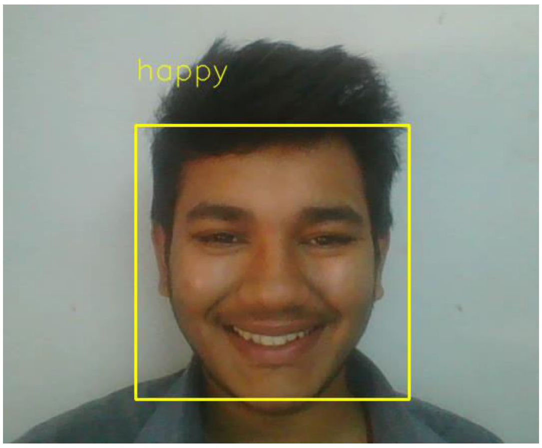

Fig.7.2.Happy

The six basic emotions and neutral as discussed , with each image representing each of those emotions(Fig.7.1-7.7).

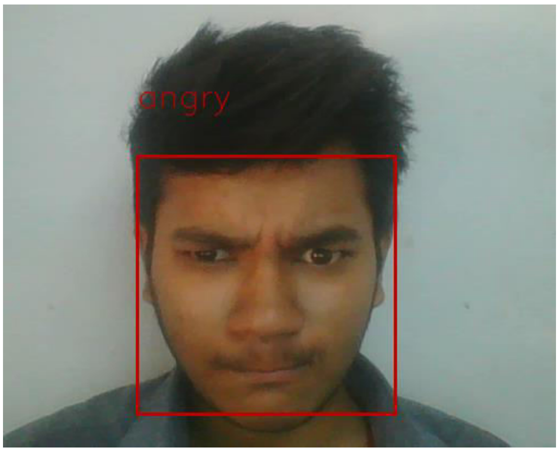

Fig.7.3.Angry

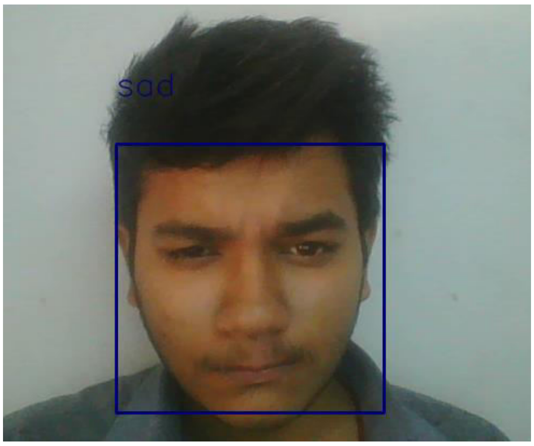

Fig.7.4.Sad

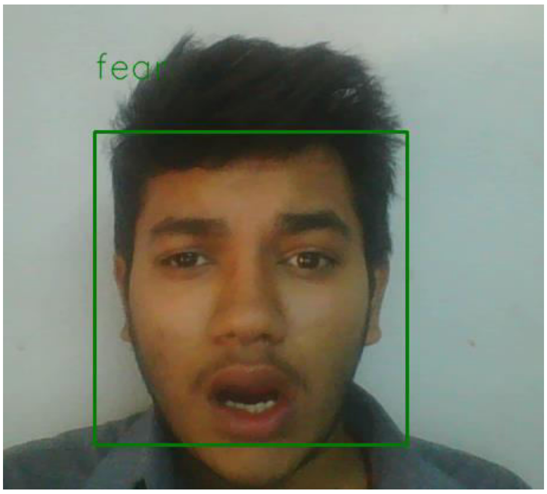

Fig.7.5. Fear

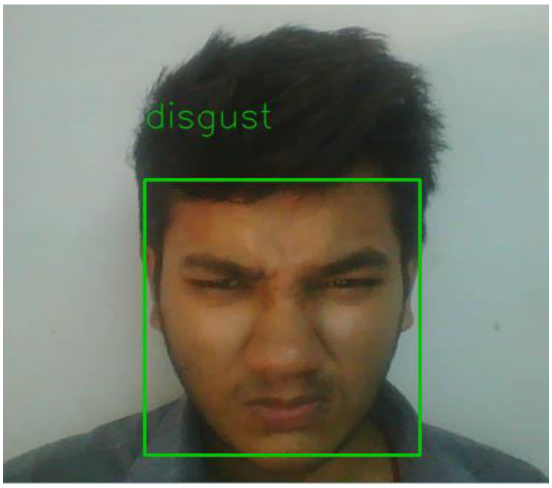

Fig.7.6.Disgust

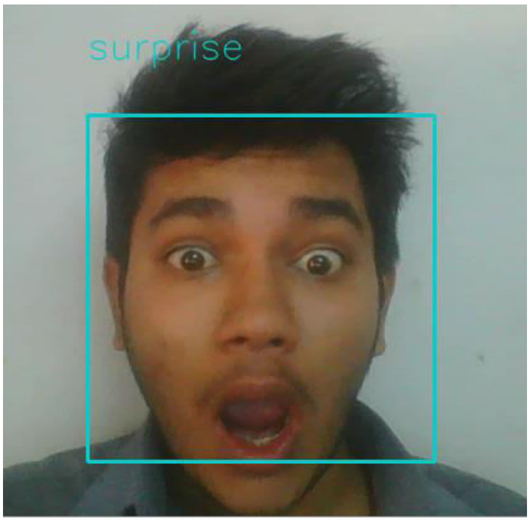

Fig.7.7.Surprise

7.2. Integration Testing

Integration testing is a logical extension of unit testing . Integration testing is defined as the testing of combined parts of an application to determine if they function correctly . Integration testing identifies problems that occur when units are combined . By using a test plan that requires you to test each unit and ensure the viability of each before combining units , you know that any errors discovered when combining units are likely related to the interface between units . This method reduces the number of possibilities to a far simpler level of analysis.

Integration of the above code onto raspberry pi without errors(integration testing)

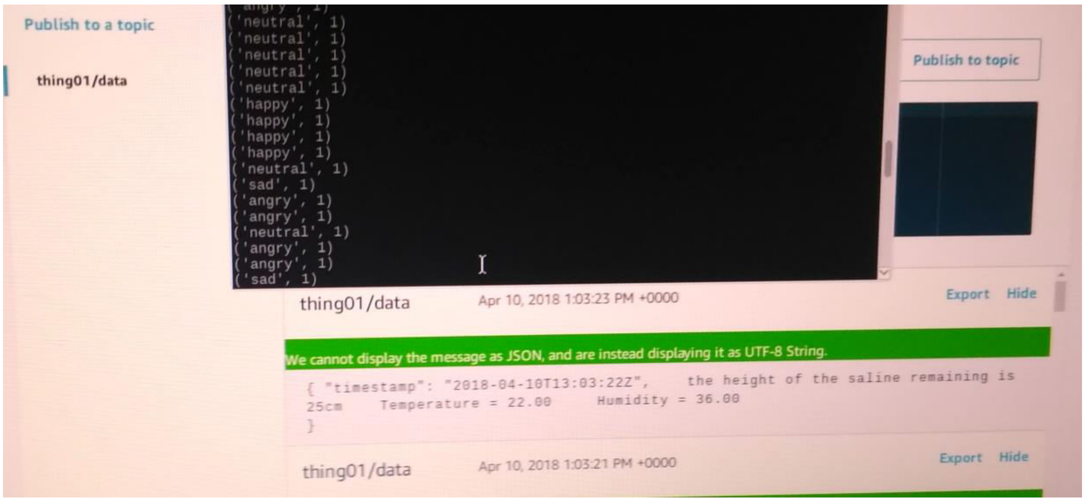

Fig.7.8.Results of sensor module on web service that is aws along with emotions relay on the monitor.

7.3. System Testing

System testing tests the system as a whole . Once all the components are integrated , the application as a whole is tested rigorously to see that it meets the specified Quality Standards . System testing is the first step in the Software

Development Life Cycle , where the application is tested as a whole. The application is tested thoroughly to verify that it meets the functional and technical specifications. System testing enables us to test, verify, and validate . It falls within the scope of black-box testing.

Fig.7.8 Shows the testing results of the entire system(System Testing)

CHAPTER 8

IMPLEMENTATION

Given in Fig.4.3 is the sequence of steps on how the three module performed with the workflow model . Sensor module with central unit arduino along with the list of sensors, the connections where made . Each sensor was connected to its designated pins along with connections to the voltage and the ground . All analog sensors are connected to the analog pins while digital ones on the digital pins of the arduino . The arduino must be coded to read sensors based on the type of sensor . After reading of data from the surroundings the arduino compares the data with the threshold displays on the serial monitor of the arduino . The task of sensor module is to relay real time data on the serial port so that raspberry pi can read it.

A series of steps takes place while finding the emotion of the inputted face of the patient . The capture of video was done by the picamera as mentioned above whose library was installed on the embedded device which its connected to the raspberry pi.
By reducing the frame rate a faster speed of detection was possible . The resolution of the camera was set to 640*480 for similar reasons . Having set u the hardware and needed inputs a python code was executed with all imported libraries whose versions and names have been given in the requirements.

Before checking for the emotions of the patient a model was developed which is trained to read emotions based on the face . As given here a python code is executed to train the model . The prerequisites for training the model is a dataset which has faces of peoples emotion from neutral with all the six basic emotions one after the other . The data sets like CK+,FER-2013 Faces Database,RaFD can be found online . For the system the FER database was used which has peoples faces, for each of the emotions that are (neutral,happy,sad,surprise,disgust,angry and fear) . The training of the model took a lot of time, and training can be done on a personal computer not necessarily on the pi . After the training is completed the model was imported onto the pi and the path to the model given to the code that ran the below script.

The camera reads the input video and divides it into an array of frames , which are formed per the second . Each split frame of the video would be converted to a greyscale image where each pixel is represented by the amount of light ,that is, it carries only intensity information . After converting the image the most important part was to find the faces in each image . For finding faces a cascade classifier(XML file) was used which finds faces in the greyscale image and write it to a variable . The variable containing all the faces are resized for the model trained to take as an input . The resized images of faces are sent to the model which predicts the possibility of each emotion and the emotion that has the highest possibility is selected and displayed . The above is done for all the faces found in the frame .

Each frame was saved to the pirgbarray which is a buffer that holds the frame input . So before moving to the next frame the buffer should be cleared.

To display the results of each module a web service was used , for that a python code which establishes connection with Amazon web service was executed and data is pushed onto the server for display and storage . The procedure for connecting and displaying data was given by Amazon in its documentation on AWS IoT.

8.1 Running the system

8.1.1 Emotion Recognition

Move to emotions directory which has models , utils library and emotion.py

Execute by- **python emotion.py**

8.1.2 Sensor and Web Service modules

After uploading the arduino with the necessary sensor code.

Attach the arduino to raspberry pi through serial communication and run **publish.py** after downloading the necessary certificates.

8.2 Problems Faced:

1.A minimum of 16 gb Memory card for installing raspian is necessary(cards with less memory will not do the work).

2.The versions above specified are very important (as syntax may change with versions)

3.Arduino must be handled carefully as they can be prone to short circuit.

4.Memory Cards on raspberry pi are sensitive can get crashed.

5. Connecting the sensors to the arduino must be done correctly to avoid false readings.

CHAPTER 9

CONCLUSION AND FUTURE PLANS

The work above discussed the use of images and sensors to monitor the health of the patients who might be at home or hospital and offer them health care with IoT infrastructure . The system detects a face 99% of the time while giving the emotion of the person with around 70% accuracy . So each patient can have their own monitoring system which can be looked after by anyone with the authorization . While sensors provide information of the patient vital and the emotion recognizer detects emotion , the care taker can predict pain if any based on the relay of the emotions detected . Future works would be to , implement a data mining concept that uses the emotion data to check for series of emotion for detecting pain . The advantages of the recommended system would be that , both emotion recognizer and sensor module that works together and if one of them fails the other can take up the load of monitoring the patient until the failed unit is back and running . Sleep patterns can also be monitored using the system . One of down side of the system would be a single point of failure that is raspberry pi where all the data is accumulated.While the data previously generated can be protected by using cloud database which is separated from the pi.Any corruption in the pi may lead to a system failure . As research grows there can be methods to increase accuracy and speed of the emotion recognize .

REFERENCES

[1] **Leandro Y. Mano , Bruno S. Faiçal , Luis H.V. Nakamura , b , Pedro H. Gomes ,∗ ,Giampaolo L. Libralon , Rodolfo I. Meneguete , Geraldo P.R. Filho , Gabriel T. Giancristofaro , Gustavo Pessin , Bhaskar Krishnamachari , JóUeyama,(2016),** from Exploiting IoT technologies for enhancing Health Smart Homes through patient identification and emotion recognition.

[2]**Paweł Tarnowski, Marcin Kołodziej, Andrzej Majkowski, Remigiusz J. Rak ,(2017)**,Emotion recognition using facial expressions.

[3]**Giampaolo L. Libralon,Roseli A. F. Romero,(2015)**,Mapping of Facial Elements for Emotion Analysis.

[4]**Ashwini Ann Varghese,Jacob P Cherian,Dr. Jubilant J Kizhakkethottam,(2016)**,Overview on Emotion Recognition System.

[5] **T. Mitchell** , Machine Learning, McGraw Hill, 1997 .

[6]**Stan Z. Li ,Anil K. Jain** ,Handbook of Face Recognition.

[7] **Husam Salih,Lalit kulkarni,(2017)**,Study of Video based Facial Expression and Emotions Recognition Methods.

[8] **P. Lucey , J.F. Cohn , et al. ,(2010)**, The extended Cohn-Kanade dataset (CK+): a complete dataset for action unit and emotion-specified expression, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops.

[9]**J. Russel**l,(1980) , A circumplex model of affect, J. Personality Soc. Psychol.

[10]**Himadri Nath Saha, Supratim Auddy, Subrata Pal, Shubham Kumar,Shivesh Pandey, Rocky Singh, Amrendra,Kumar Singh, Priyanshu Sharan, Debmalya Ghosh, Sanhita Saha,(2015)**,Health Monitoring and Management Using Internet-of-Things (IoT) Sensing with Cloud-Based Processing: Opportunities and Challenges.

[11]**Pratiksha W. Digarse,Sanjaykumar L.Patil**,(2017),Arduino UNO and GSM Based Wireless Health Monitoring System for Patients.

[12]**Andr´e Teixeira Lopes, Edilson de Aguiar, Alberto F De Souza, and Thiago Oliveira-Santos,(2016)**. Facial expression recognition with Convolutional Neural Networks.

[13]**Mostafa Mohammadpour , Hossein Khaliliardali , Seyyed Mohammad R Hashemi, Mohammad . M AlyanNezhadi,(2016)**,Facial Emotion Recognition using Deep Convolutional Networks

[14]**Enrique Correa ,Arnoud Jonker ,Micha□el Ozo ,Rob Stolk,(2017)**,Emotion Recognition using Deep Convolutional Neural Networks.

[15]**Dhiraj Sunehra,Pini Ramakrishna,(2016)**, Web based patient health monitoring system using Raspberry Pi.

[16]**Mohan Lal Sahu, Jigyasu Kumar Kaushal,** Real time health monitoring system using Arduino and LabVIEW with GSM Technology.

[17] **T.Saraswathi, S.Amutha,(2016)**, Iot Based Wireless Healthcare Monitoring.