DESIGN AND FABRICATION OF BIOMETRIC FACE RECOGNITION SECURITY SYSTEM

MT6713 DESIGN AND FABRICATION LAB

A PROJECT REPORT

Submitted by

VISHAL CHANDRU 211616115057

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

MECHARONICS ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, THANDALAM.
ANNA UNIVERSITY: CHENNAI 600025
October 2019

PROGRAMME EDUCATIONAL OBJECTIVES

- Graduates will have comprehensive knowledge in the analytical, scientific and engineering fundamentals necessary to model, analyse and solve engineering problems and to prepare them for graduate studies and for successful careers in industry.
- Graduates will effectively design and develop products in the areas such as manufacturing, motion control, machine vision, system simulation, intelligent systems, automotive systems and robotics.
- Graduates will acquire Technical expertise, Leadership skills, Ethical practices and Team spirit with a concern towards greener society.

PROGRAMME OUTCOMES

- **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- **Problem analysis:** Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
- **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

- **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES

- To innovate a Mechatronics system to meet the requirements and specifications.
- To analyse and improve the performance of a Mechatronics system and enhance the intellectual capabilities of the system.
- To lead professional career in industries or an entrepreneur by applying Engineering and Management principles and practices

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report titled **"DESIGN AND FABRICATION OF BIOMETRIC FACE RECOGNITION SECURITY SYSTEM"** is the bonafide work of "**AJITH KEVIN ANAND.K(211616115002)**, **ILAMPARITHI.K(211616115021)** and **VISHAL.C(211616115057)"** who carried out the project under my supervision.

SIGNATURE

Dr. V. SANTHANAM

Professor & HEAD

Department of Mechatronics Engg

Rajalakshmi Engineering College,

Thandalam, Chennai-602 105.

SIGNATURE

Mr. I. LEANDO

Assistant professor

Department of Mechatronics Engg

Rajalakshmi Engineering college,

Thandalam, Chennai-602 105.

Submitted for the practical examination held on …….

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

Initially we thank the almighty for being with us through every walk of life it is our privilege to express our sincerest thanks to our respected Chairman **Mr. S. Meganathan**, and beloved Chairperson **Dr.Thangam Meganathan**, for providing us with the requisite infrastructure and extending support in all endeavors.

Our heartfelt thanks to Dr. S. N. Murugesan, our Principal for his kind support and resources provided to complete our work in time. We also thank **Dr. G. Thanigaiyarasu**, Dean Mechanical Sciences for his suggestion and guidance for completion of project.

We deeply express our sincere thanks to **Dr. V. Santhanam**. Head of our Department, for his encouragement and continues support to complete the project in time.

We are glad to express our sincere indebtedness to our project coordinators **Dr. M. Balakarthikeyan**, Assistant professor, Department of Mechatronics Engineering for their constructive criticism throughout the duration of our project.

We are glad to express our sincere thanks and regards to our supervisor **Mr. I. Leando**, Assistant Professor, Department of Mechatronics Engineering for his guidance and suggestion throughout the course of the project.

Finally, we express our thanks for all teaching, non-teaching faculty of our Mechatronics Engineering department for helping us with the necessary suggestions and guidance during the time of project.

ABSTRACT

We present an approach to secure homes using facial recognition technology combined with the existing locking mechanism to improve the security with the motive of cost efficiency. Our approach uses sub-machine vector to identify the face and capture a two-dimensional image by taking specific points on the face and comparing it with the stored images. Our major goal is to improve security using software that is everyday being developed for the better. Our motive is to produce a highly secure biometric system for the replacement of physical locks to reduce the amount of theft and creating a safer environment. It is also our objective to achieve them by a cost effective way so that is reaches every people from different walks of life. What we have achieved is to be believed that it can revolutionize the existing generation of lock mechanism and a hands free secure system is also achieved.9

INDEX

CHAPTER-1

INTRODUCTION

The world that we live in has improved drastically with the recent technological advancements and most of these technological advancements have been achieved because of the development in software which in turn improves the hardware.

DEEP LEARNING

Deep learning is an artificial intelligence function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or unlabelled. Also known as deep neural learning or deep neural network.

FACIAL RECOGNITION

Facial recognition is a biometric software application capable of uniquely identifying or verifying a person by comparing and analysing patterns based on the person's facial contours. Facial recognition is mostly used for security purposes, though there is increasing interest in other areas of use. In fact, facial recognition technology has received significant attention as it has potential for a wide range of application related to law enforcement as well as other enterprises

As the need for security is becoming mandatory, facial recognition has slowly become one of the most secure platforms for protection from threats. It uses deep learning or machine learning techniques to identify the characteristics of a

person’s face. This technology became a huge success in the smartphone industry and now we have implemented this technology in homes for unlocking doors.

India burglary report per 1,00,000 population

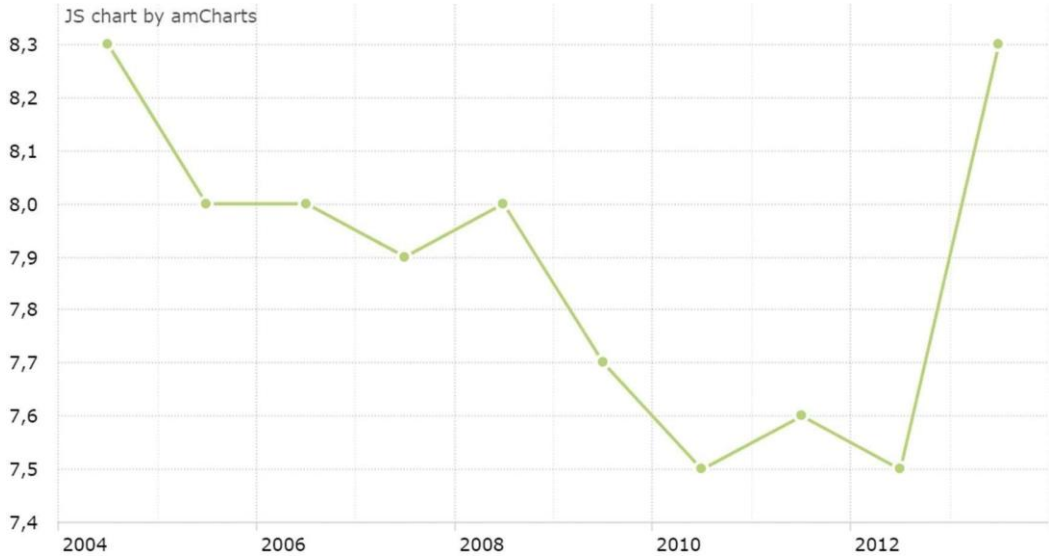


FIGURE 1.1

Growth in burglary per 1,00,000 population

Data	Date	Evolution
8.30	2013	~ 0.80
7.50	2012	~ -0.10
7.60	2011	~ 0.10
7.50	2010	~ -0.20

Ranking of the country (India) at the global level is (from the highest to the lowest data) : 99 / 108

FIGURE 1.2

Average rate of burglary

Data	Date of information
8.30	2013
7.50	2012
7.60	2011
7.50	2010
7.70	2009
8.00	2008
7.90	2007
8.00	2006
8.00	2005
8.30	2004

FIGURE 1.3

CHAPTER-2

LITERATURE REVIEW

2.1 TITLE: Smart digital door lock for home automation

AUTHORS: Yong Tae Park, Pranesh Sthapit

ABSTRACT:

In this paper, we propose a smart digital door lock system for home automation. A digital door lock system is equipment that uses the digital information such as a secret code, semi-conductors, smart card, and fingerprints as the method for authentication instead of the legacy key system. In our proposed system, a ZigBee module is embedded in digital door lock and the door lock acts as a central main controller of the overall home automation system. Technically, our proposed system is the network of sensor nodes and actuators with digital door lock as base station. A door lock system proposed here consists of RFID reader for user authentication, touch LCD, motor module for opening and closing of the door, sensor modules for detecting the condition inside the house, communication module, and control module for controlling other modules. Sensor nodes for environment sensing are deployed at appropriate places at home. Status of individual ZigBee module can be monitored and controlled by the centralized controller, digital door lock. As the door lock is the first and last thing people come across in entering and leaving the home respectively, the home automation function in digital door lock system enables user to conveniently control and monitor home environment and condition all at once before entering or leaving the house. Furthermore, it also allows users to remotely monitor the condition inside the house through Internet or any other public network. The biggest advantage of our proposed system over existing ones is that it can be easily installed when and where necessary without requirement of any infrastructures and proper planning.

Published on 23rd January 2009 in IEEE

2.2 TITLE: Smart door lock

AUTHOR: JF Buzhardt

ABSTARCT:

A self-contained door locking apparatus is disclosed which collects, stores, displays, and/or transmits information each time the apparatus is opened, closed, or even merely handled. This information may include but is not limited to time, date, key ID, and the number of times the apparatus is used or handled. The functions and performance of the locking apparatus may be manually or remotely controlled and manipulated. The disclosed apparatus may be also manually or remotely interrogated and the information gathered by theapparatus may be locally stored and/or be transmitted to a remote receiver such as a cell phone or a computing device.

Published on 2015

2.3 TITLE: Electronically activated door lock

AUTHOR: C. Doucet

ABSTACT:

A door lock assembly which includes an electronic card reader is provided whereby when the clutch assembly is in the activated position, rotational movement of the outside handle is transmitted by way of a spindle through the mortise and through the clutch assembly to a clutch disk disposed between the mortise and clutch assemblies and the inside housing. The clutch disk is disposed inside the door. Rotational movement of the clutch disk is then imparted back

towards the door through a drive disk and hub drive to the mortise latch hub which is disposed in the mortise housing disposed inside the door structure. As a result, the clutch disk and clutch assembly can be disposed between the inside housing and the inside door surface and the electronic components of the card reader and clutch assembly can be conveniently housed in the inside housing assembly. An additional spacer hub is disposed between the outside housing and the mortise latch hub which makes it difficult to tamper with the mortise latch hub from the outside of the door.

Published on 2000

2.4 TITLE: Face Recognition using Eigenfaces

AUTHORS: Mathew A. Turk, Alex P. Pentland

ABSTRACT:

We present an approach to the detection and identification of human faces and describe a working, near-real-time face recognition system which tracks a subject's head and then recognizes the person by comparing characteristics of the face to those of known individuals. Our approach treats face recognition as a two-dimensional recognition problem, taking advantage of the fact that faces are normally upright and thus may be described by a small set of 2-D characteristic views. Face images are projected onto a feature space ("face space") that best encodes the variation among known face images. The face space is defined by the "eigenfaces", which are the eigenvectors of the set of faces; they do not necessarily correspond to isolated features such as eyes, ears, and noses. The framework provides the ability to learn to recognize new faces in an unsupervised manner.

Published on 1999

2.5 TITLE: Comparative Analysis of RFID and Wireless Home/Office Automation

AUTHORS: Manasee Patil, S.R.N. Reddy

ABSTRACT:

Wireless Sensor Network (WSN) is most widely used wireless technology in different applications. Home automation makes day to day life of people easier. WSN provides flexible management of lighting, heating, cooling and security from anywhere in the home/office [20]. In this project we propose use of both wired and wireless technology for home/ office automation. RFID technology is used for automatic door opening & closing. We also propose use of wireless sensor network for temperature, lighting, smoke detection and automatic door opening & closing. GSM technology is used in this project to monitor and control various devices from outside the home/office.

Published on 2013

2.6 TITLE: Smart digital door lock system using Bluetooth technology

AUTHORS: Siddhi Kavade, Riddhi Kavade

ABSTRACT:

Today's world is the smart world. In this smart world, everything that is from home to city everything is smart. So, in this paper part of the smart home is given that is smart door lock system. In this owner can control their door using the smartphone by viewing the live feed. One database is there for storing visitor's information. By using Bluetooth technology this all system works.

Published on 2017

2.7 TITLE: Deep Face Recognition

AUTHORS: Omkar M. Parkhi

ABSTRACT:

The goal of this paper is face recognition – from either a single photograph or from a set of faces tracked in a video. Recent progress in this area has been due to two factors: (i) end to end learning for the task using a convolutional neural network (CNN), and (ii) the availability of very large-scale training datasets. We make two contributions: first, we show how a very large scale dataset (2.6M images, over 2.6K people) can be assembled by a combination of automation and human in the loop, and discuss the trade-off between data purity and time; second, we traverse through the complexities of deep network training and face recognition to present methods and proceduresto achieve comparable state of the art results on the standard LFW and YTF face benchmarks.

Published on 2015

2.8 TITLE: Face recognition by elastic bunch graph matching

AUTHORS: Jean-Marc Fellous, Norbert Krüger, Christoph von der Malsburg, Laurenz Wiskott

ABSTRACT:

We present a system for recognizing human faces from single images out of a large database with one image per person. The task is difficult because of image variation in terms of position, size, expression, and pose. The

system collapses most of this variance by extracting concise face descriptions in the form of image graphs. In these, fiducial points on the face (eyes, mouth etc.) are described by sets of wavelet components (jets). Image graph extraction is based on a novel approach, the bunch graph, which is constructed from a small set of sample image graphs. Recognition is based on a straight-forward comparison of image graphs. We report recognition experiments on the FERET database and the Bochum database, including recognition across pose

Published on 1997

2.9 TITLE: Face Recognition with Local Binary Patterns

AUTHORS: Timo Ahonen, Abdenour Hadid, Matti Pietikäinen

ABSTRACT:

In this work, we present a novel approach to face recognition which considers both shape and texture information to represent face images. The face area is first divided into small regions from which Local Binary Pattern (LBP) histograms are extracted and concatenated into a single, spatially enhanced feature histogram efficiently representing the face image. The recognition is performed using a nearest neighbour classifier in the computed feature space with Chi square as a dissimilarity measure. Extensive experiments clearly show the superiority of the proposed scheme over all considered methods (PCA, Bayesian Intra/extra personal Classifier and Elastic Bunch Graph Matching) on FERET tests which include testing the robustness of the method against different facial expressions, lighting and aging of the subjects. In addition to its efficiency, the simplicity of the proposed method allows for very fast feature extraction.

Published on 2004

2.10 TITLE: Multi-digit electrical door lock

AUTHORS: E. J. LEONARD

ABSTRACT:

This invention relates to a multi-digit electrical door lock. lts principal object is to provide a multi-digit electrical door lock of great security against the door being unlocked by anyone other than an authorized person who knows the currently-assigned multi-digit unlocking number.

It is generally conceded that multi-digit combination locks are more secure against being opened by 'unauthorized persons than are the usual key-operated locks. The usual combination locks, however, are not entirely satisfactory where great security is important. Among other reasons, changing the combination, as when it is suspected that unauthorized persons may have learned the combination or when previously authorized persons are no longer authorized, is usually a time-consuming operation which often requires the services of a trained locksmith, wherefore the combination is frequently left unchanged too long.

Published on 1962

2.11 TITLE: Electronic door lock

AUTHORS: KW Gartner, AK Uyeda

ABSTRACT: An electronic door lock is provided with a single control knob used for entering a predetermined combination through manipulation of the knob in a first arc of rotation, the code being entered by pushing the dial inwardly to bring a push pad into contact with individual ones of an arcuate array of electrical

switches provided on a printed circuit board within the lock housing, the release of the door locking bolt being accomplished after entry of the predetermined code by further manipulation of the control knob through remaining portions of knob rotation which are unavailable until after entry of the predetermined code.

Published on 1990

CHAPTER-3

FABRICATION PROCESS

MATERIALS:

- **FABRICATION**
  - Plywood (3*2.5 Feet)
  - Nails (3\4 inch)
  - Nut and Bolts
  - Hinge 2 pieces
  - Spur gear
  - Rack
  - U – Clamps (3 pieces)
- **DESIGN SOFTWARE:**
  - AUTOCAD

ELECTRICAL COMPONENETS:

- **Servo Motor**
  - Model: MG995
  - Weight: 55 gm
  - Operating voltage: 4.8V~ 7.2V
  - Servo Plug: JR
  - Stall torque @4.8V : 10 kg-cm
  - Stall torque @6.6V : 12 kg-cm

- **Arduino Uno R3**
  - Microcontroller: ATmega328
  - Operating Voltage: 5V

- Input Voltage (recommended): 7-12V
- Clock Speed: 16 MHz
- DC Current per I/O Pin: 40 mA

♦ **Power Distribution Board Version 2**

CIRCUIT

- Arduino UNO R3
- Servo motor
- Power Distribution Board
- Jumper Wires

SOFTWARE:

- Python 3.7 IDE
- IP Webcam App
- Arduino

3.1 PLYWOOD

FIGURE 3.1

Plywood is a material manufactured from thin layers or "plies" of wood veneer that are glued together with adjacent layers having their wood grain rotated up to 90 degrees to one another. It is an engineered wood from the family of manufactured boards which includes medium-density fibreboard (MDF) and particle board (chipboard).

All plywood bind resin and wood fibre sheets (cellulose cells are long, strong and thin) to form a composite material. This alternation of the grain is called cross-graining and has several important benefits: it reduces the tendency of wood to split when nailed in at the edges; it reduces expansion and shrinkage, providing improved dimensional stability and it makes the strength of the panel consistent across all directions. There is usually an odd number of plies, so that the sheet is balanced-this reduces warping. Because plywood is

banded with grains running against one another and with an odd number of composite parts, it has high stiffness perpendicular to the grain direction of the surface ply.

3.2 HINGE

FIGURE 3.2

A hinge is a mechanical bearing that connects two solid objects, typically allowing only a limited angle of rotation between them. Two objects connected by an ideal hinge rotate relative to each other about a fixed axis of rotation: all other translations or rotations being prevented, and thus a hinge has one degree of freedom Hinges may be made of flexible material or of moving components.

3.3 SPUR GEAR

FIGURE 3.3

Spur gears or straight-cut gears are the simplest type of gear. They consist of a cylinder or disk with teeth projecting radially. Though the teeth are not straight-sided (but usually of special form to achieve a constant drive ratio, mainly involute but less commonly cycloidal), the edge of each tooth is straight and aligned parallel to the axis of rotation. These gears mesh together correctly only if fitted to parallel shafts. No axial thrust is created by the tooth loads. Spur gears are excellent at moderate speeds but tend to be noisy at high speeds.

Spur gear teeth are manufactured by either involute profile or cycloidal profile. Most of the gears are manufactured by involute profile with 20° pressure angle. When two gears are in mesh at one instant there is a chance to mate involute portion with non-involute portion of mating gear. This phenomenon is known as "interference" and occurs when the number of teeth on the smaller of the two meshing gears is less than a required minimum. To avoid interference we can have undercutting, but this is not a suitable solution as undercutting leads to weakening of tooth at its base. In this situation Corrected gears are used. In corrected gears Cutter rack is shifted upwards or downwards.

3.4 RACK

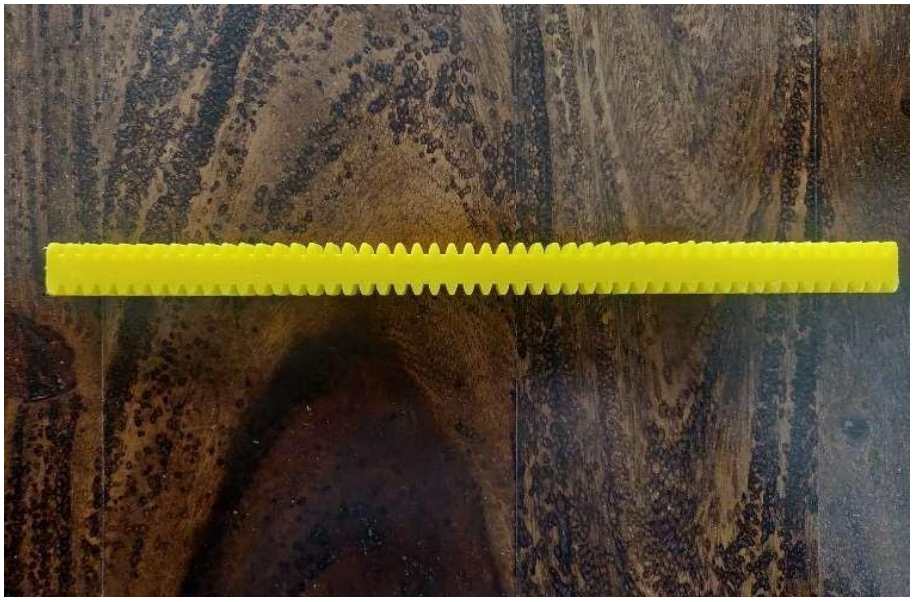


FIGURE 3.4

Gear racks are utilized to convert rotating movement into linear motion. A gear rack has straight teeth cut into one surface of a square or round section of rod and operates with a pinion, which is a small cylindrical gear meshing with the gear rack. Generally, gear rack and pinion are collectively called “rack and pinion”. There are many ways to use gears. For example, as shown in the picture, a gear is used with the gear rack to rotate a parallel shaft.

To provide many variations of rack and pinion, KHK has many types of gear racks in stock. If the application requires a long length requiring multiple gear racks in series, we have racks with the tooth forms correctly configured at the ends. These are described as “gear racks with machined ends”. When a gear rack is produced, the tooth cutting process and the heat treatment process can cause it to try & go out of true. We can control this with special presses & remedial processes.

There are applications where the gear rack is stationary, while the pinion traverses and others where the pinion rotates on a fixed axis while the gear rack moves. The former is used widely in conveying systems while the latter can be used in extrusion systems and lifting/lowering applications.

3.5 U-CLAMPS

FIGURE 3.5

A clamp is a fastening device used to hold or secure objects tightly together to prevent movement or separation through the application of inward pressure.It fastened using ½ inch screws.

3.6 SERVO MOTOR

FIGURE 3.6

it has metal gears which makes it robust and reliable motors. These TowerPro MG995 Metal Gear Servo Motors are the high-speed servo motors with the mighty torque of 10 kg/cm.

The TowerPro MG995 High-Speed Digital Servo Motor rotates 90° in each direction making it 180° servo motor. It is a digital servo motor which receives and processes PWM signal faster and better. It equips sophisticated internal circuitry that provides good torque, holding power, and faster updates in response to external forces.

They are packed within a tight sturdy plastic case which makes them water and dust resistant which is a very useful feature in RC planes, Boats, and RC Monster Trucks etc. It equips 3-wire JR servo plug which is compatible with Futaba connectors too.

Wire Description

Red - Positive

Brown - Negative

Orange - Signal

3.7 ARDUINO UNO R3

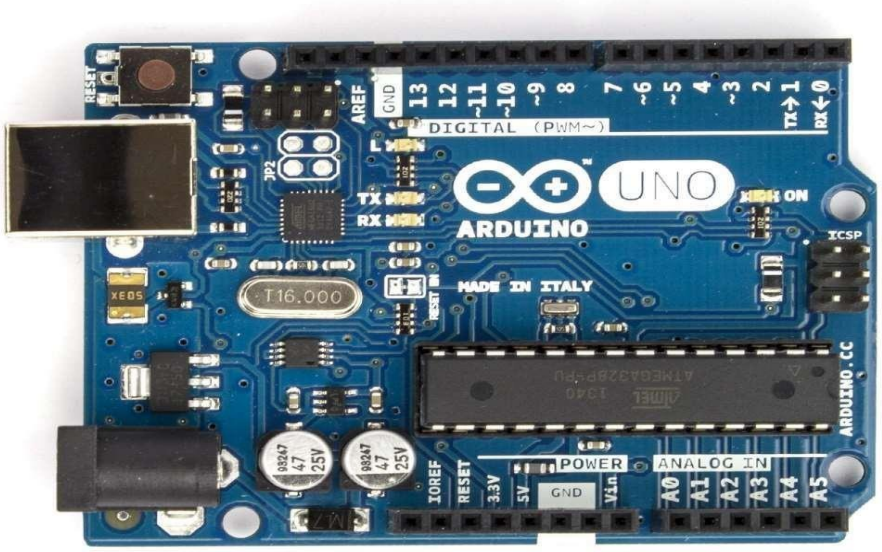


FIGURE 3.7

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference

model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards. You can find here your board warranty information

3.8 POWER DISTRIBUTION BOARD VER-2

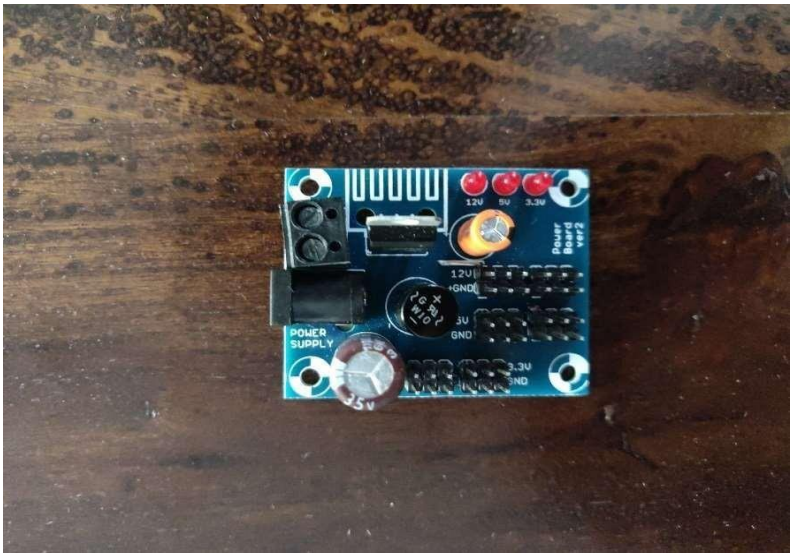


FIGURE 3.8

Power Distribution Board (PDB) is a printed circuit board that is used to distribute the power from your flight battery to all different components of the multirotor. Prior to PDB's becoming common it was necessary to connect all the different components using wire and the result often resembled an octopus and weighed a considerable amount due to the amount of copper and solder joints in the wires. There are many different PDB's available from various manufacturers, the majority of them provide very similar features. Initially PDB's were very simple and were just a thick copper PCB with an input and multiple outputs. As the need for regulated voltages for various components has become more common,

manufacturers have begun including voltage regulators on the PDB so that voltage sensitive components can be fed reliable, stable, and clean power.

3.9 JUMPER WIRES

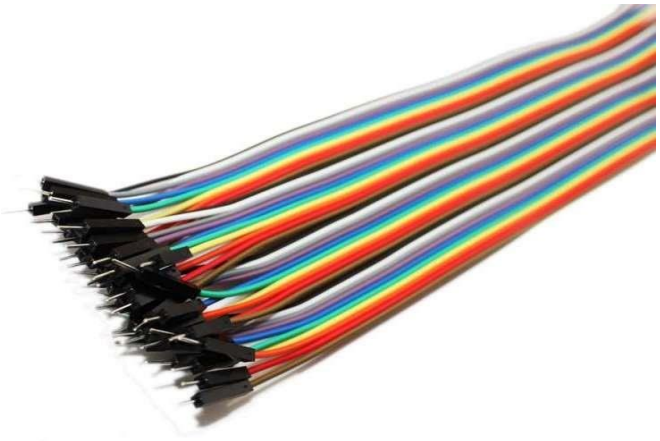


FIGURE 3.9

A jump wire (also known as jumper wire, or jumper) is an electrical wire, or group them in a cable, with a connector or pin at each end (or sometimes without them - simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

CHAPTER-4

PROTOTYPE PHOTO

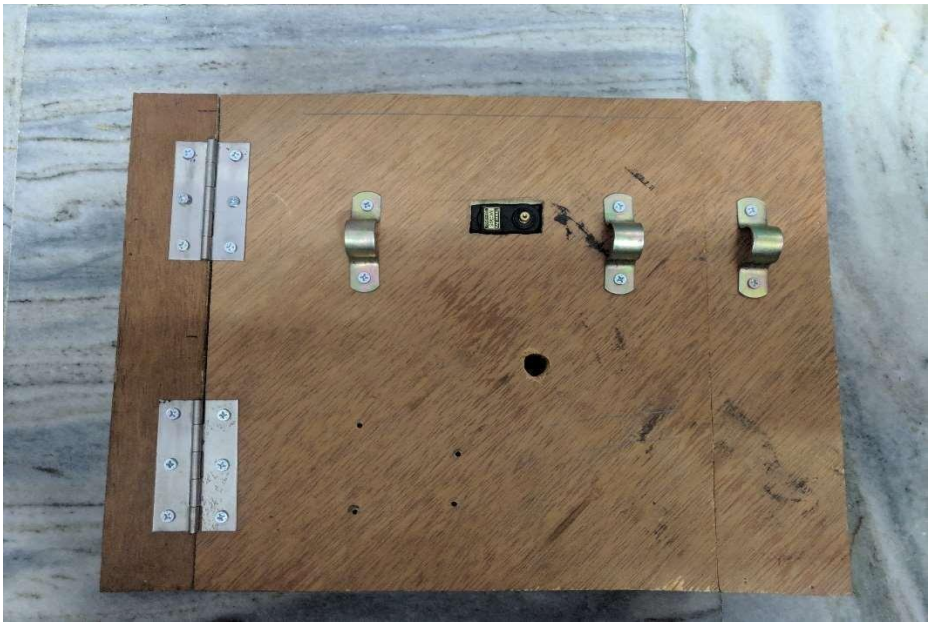


FIGURE 4.1

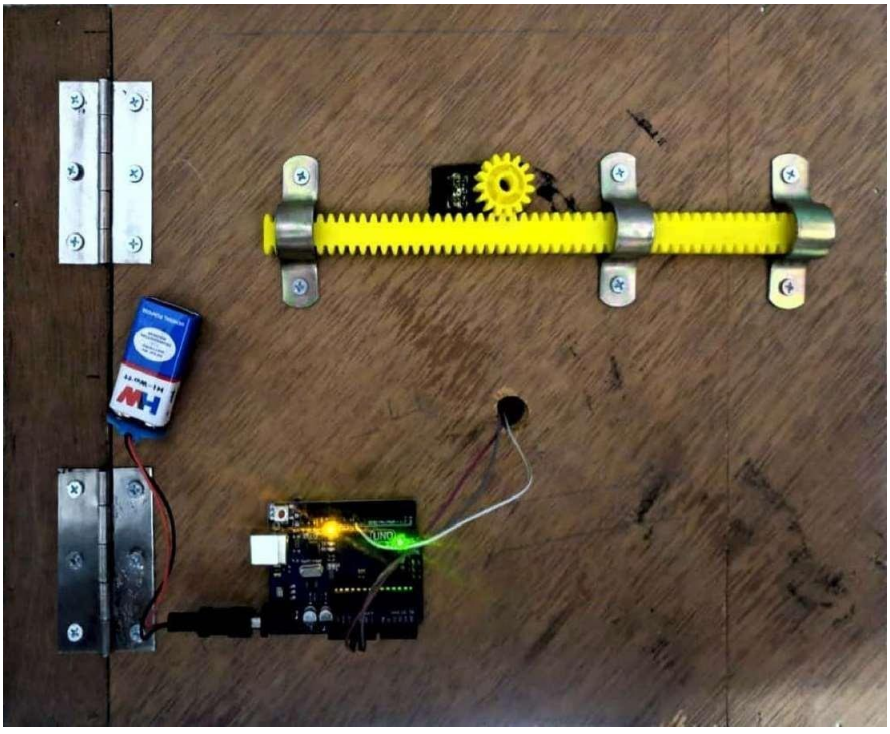


FIGURE 4.2

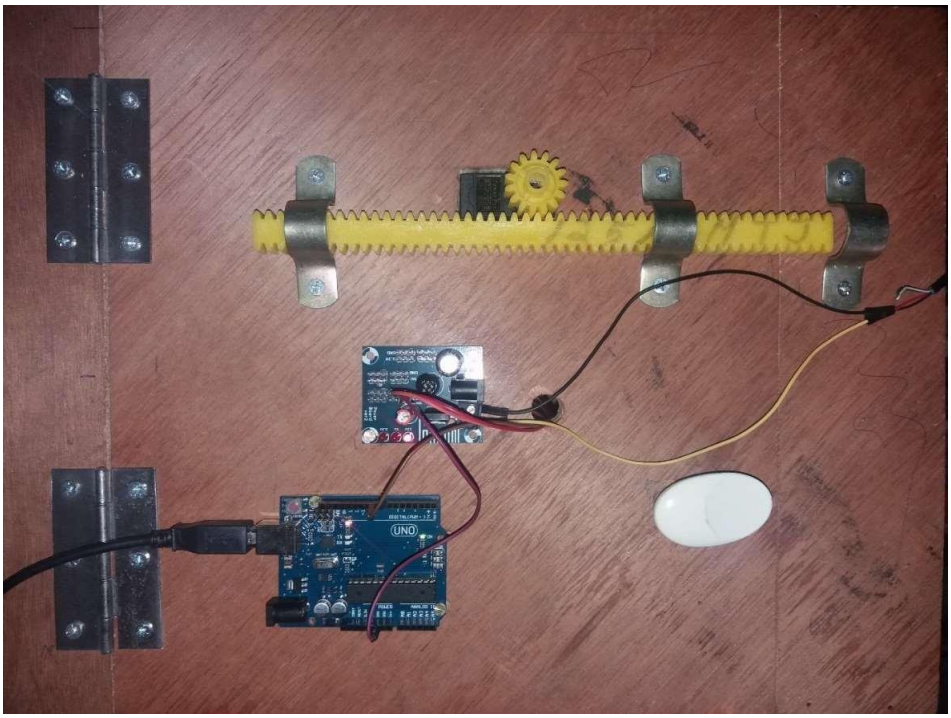


FIGURE 4.3

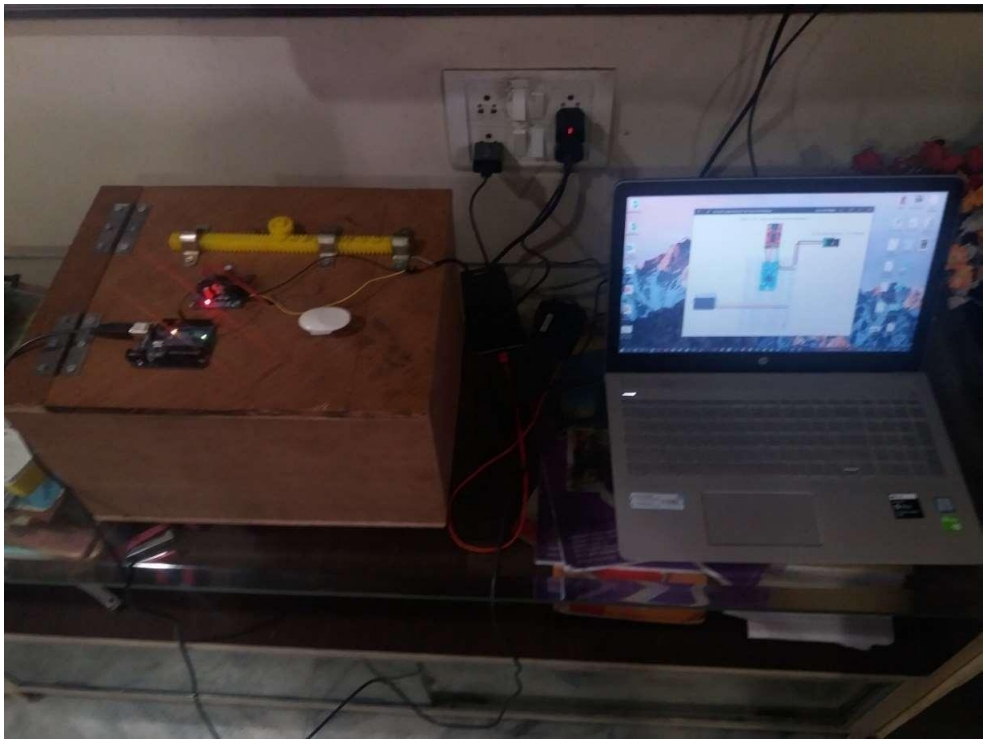


FIGURE 4.4

CHAPTER-5

DESIGN, SIMULATION AND HARDWARE

RACK AND PINION:

Rack length = 22.8cm

No. of teeth on rack = 50

Pinion diameter = 1.8cm

No. of teeth on pinion = 15

GEAR RATIO = No. of teeth on driver gear/No. of teeth on rack

= 15/50 = 3:10 = 0.3

RACK DISPLACEMENT PER 360 DEGREE ROTATION

= Gear ratio * Length of the rack

= 0.3 * 22.8

= 6.84 cm

= 3.42 cm per 180-degree rotation

Application of deep learning and OpenCV together to Detect faces:

1. Compute 128-d face embeddings to quantify a face
2. Train a Support Vector Machine (SVM) on top of the embeddings
3. Recognize faces in images and video streams
4. All of these tasks will be accomplished with OpenCV, enabling us to obtain a "pure" OpenCV face recognition pipeline.

Working of OpenCV's face recognition:

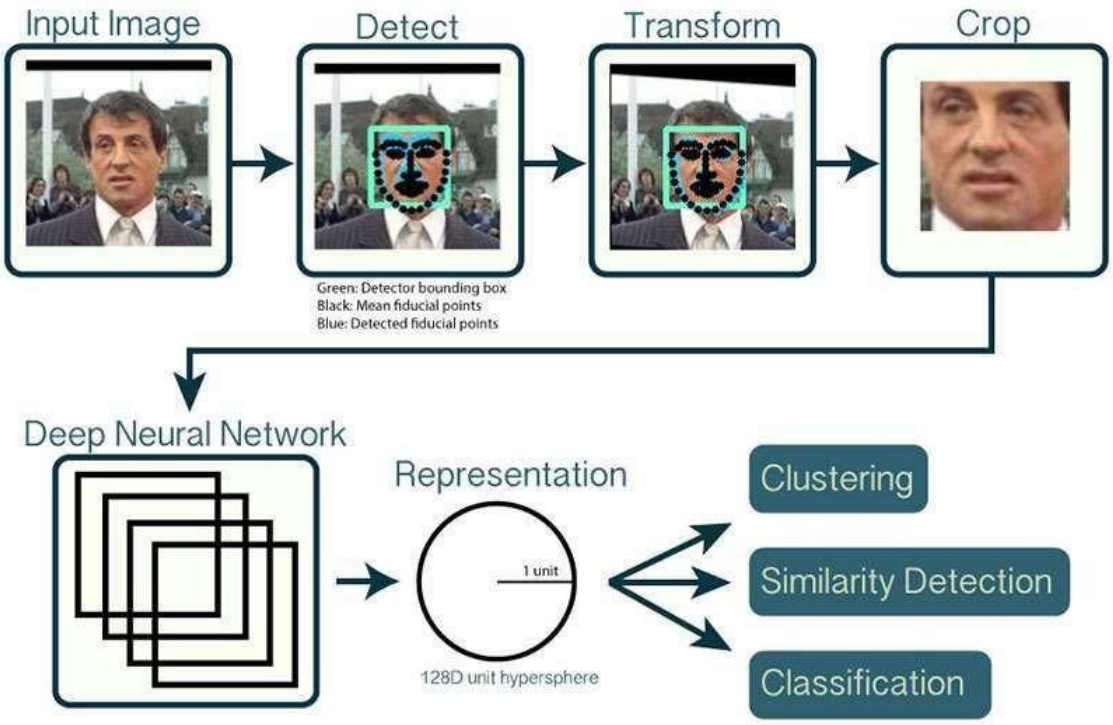


FIGURE 5.1

An overview of the OpenCV face recognition pipeline. The key step is a CNN feature extractor that generates 128-d facial embeddings.

In order to build our OpenCV face recognition pipeline, we'll be applying deep learning in two key steps:

1. To apply face detection, which detects the presence and location of a face in an image, but does not identify it
2. To extract the 128-d feature vectors (called "embeddings") that quantify each face in an image
3. The model responsible for actually quantifying each face in an image is from the OpenFace project, a Python and Torch implementation of face recognition with deep learning.
4. Reviewing the entire Face Net implementation is outside the scope of this tutorial, but the gist of the pipeline can be seen in Figure 1 above.
5. First, we input an image or video frame to our face recognition pipeline. Given the input image, we apply face detection to detect the location of a face in the image.

6. Optionally we can compute facial landmarks, enabling us to pre-process and align the face
7. Face alignment, as the name suggests, is the process of (1) identifying the geometric structure of the faces and (2) attempting to obtain a canonical alignment of the face based on translation, rotation, and scale.

8. While optional, face alignment has been demonstrated to increase face recognition accuracy in some pipelines.

9. After we've applied face alignment and cropping, we pass the input face through our deep neural network.

10. The FaceNet deep learning model computes a 128-d embedding that quantifies the face itself.

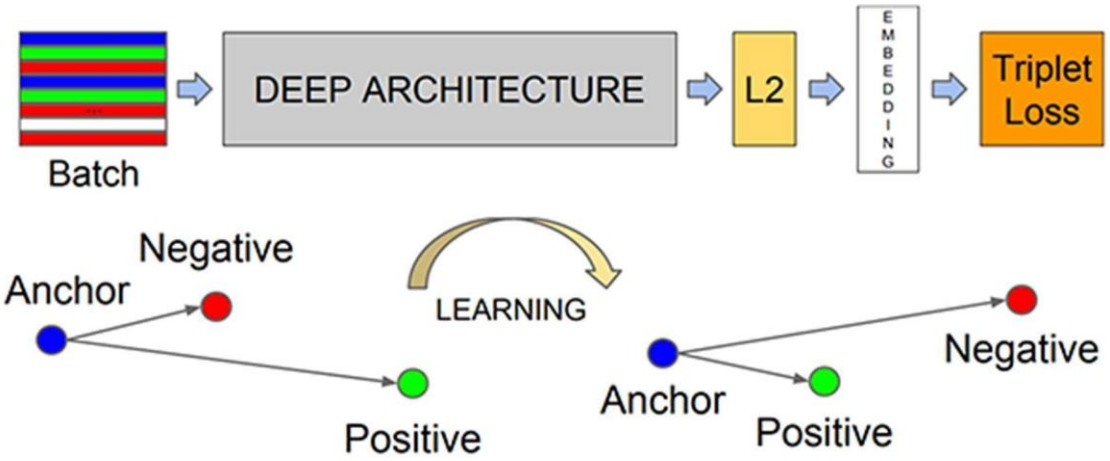


FIGURE 5.2

COMPUTATION OF FACE EMBEDINGS BY THE NETWORK

- The input data to the network
- The triplet loss function
- To train a face recognition model with deep learning, each input batch of data includes three images:
    - The anchor
    - The positive image
    - The negative image
- The anchor is our current face and has identity A.
- The second image is our positive image — this image also contains a face of person A.
- The negative image, on the other hand, does not have the same identity, and could belong to person B, C, or even Y!

- The point is that the anchor and positive image both belong to the same person/face while the negative image does not contain the same face.

- The neural network computes the 128-d embeddings for each face and then tweaks the weights of the network (via the triplet loss function) such that:

- The 128-d embeddings of the anchor and positive image lie closer together
- While at the same time, pushing the embeddings for the negative image father away
- In this manner, the network can learn to quantify faces and return highly robust and discriminating embeddings suitable for face recognition.

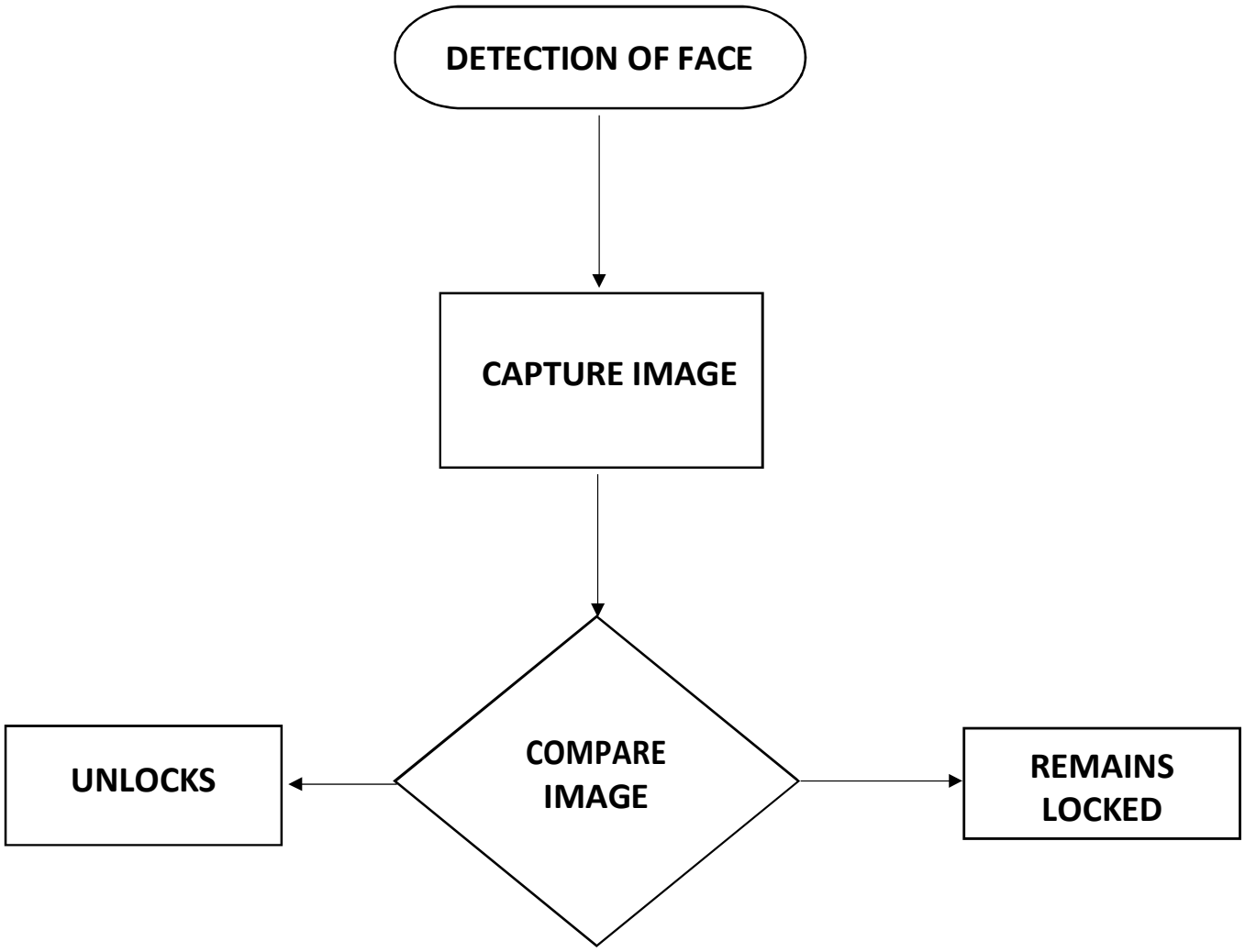
WORKING OF THE MECHANISM

- If the face captured is matched with the stored image, then a signal is sent from the Arduino to the servo motor

- Now the servo motor rotates the spur gear(pinion) which in turn moves the linear gear (rack) in clockwise direction and now the door is unlocked.

- If the image captured does not match the required percentage of success or when no image is matched, the door remains locked.

FLOW CHART DIAGRAM:



DESIGN WORK USING AUTOCAD:

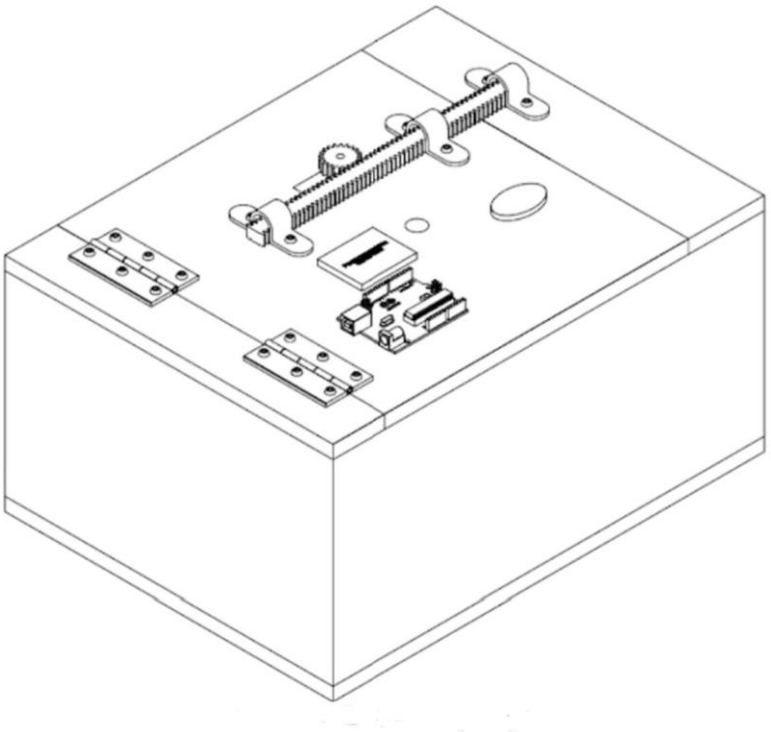


FIGURE 5.3

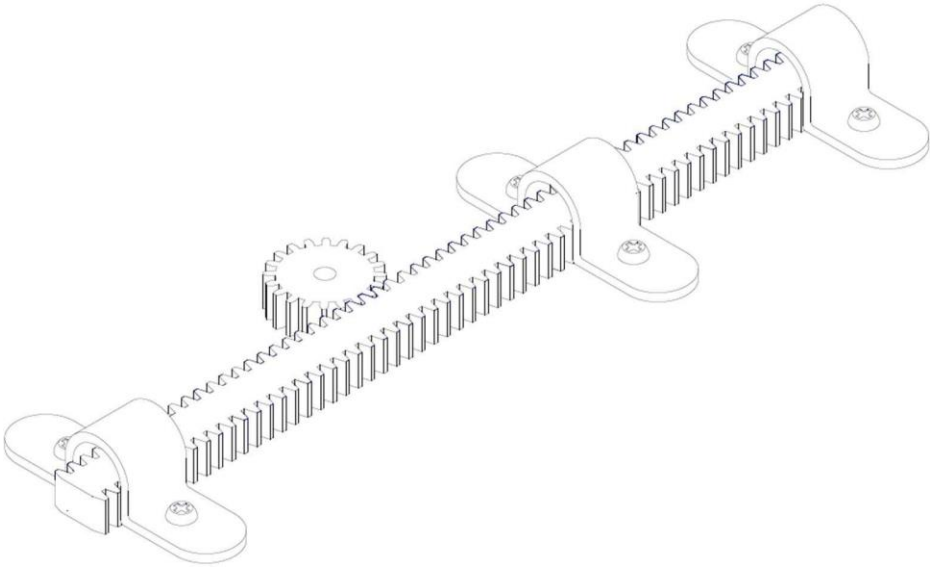


FIGURE 5.4

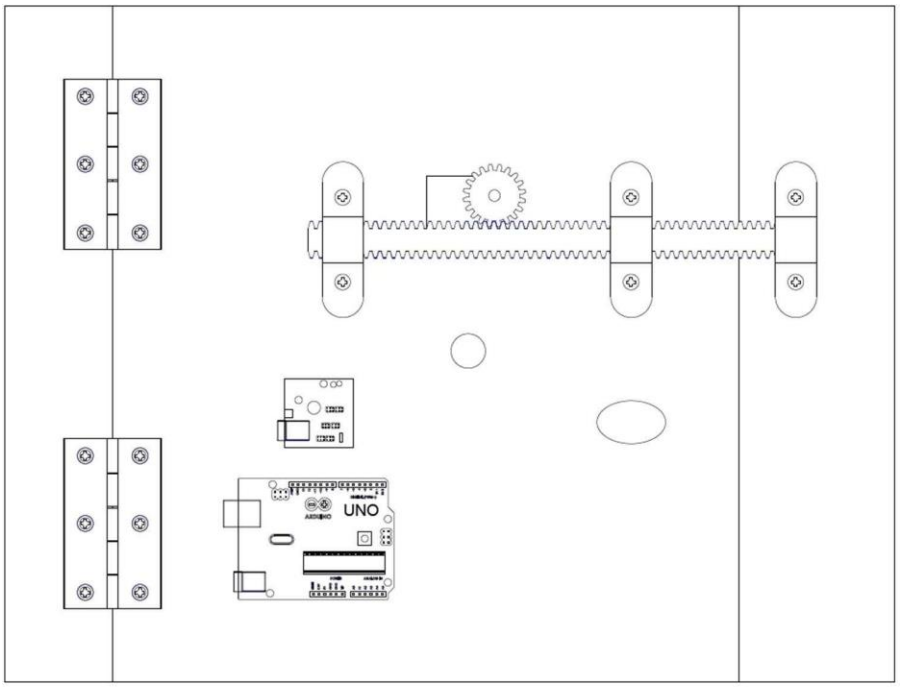


FIGURE 5.5

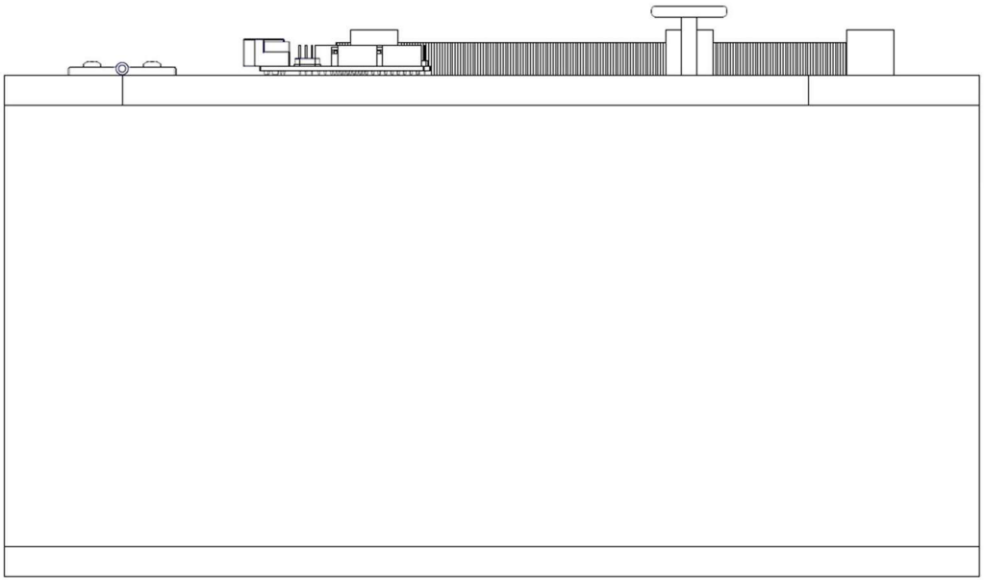


FIGURE 5.6

CODE WORK

extract_embeddings.py

```
# USAGE
# python extract_embeddings.py --dataset dataset --embeddings
output/embeddings.pickle  --detector face_detection_model --embedding-
model openface_nn4.small2.v1.t7


# import the necessary packages
from imutils import paths
import numpy as np
import argparse
import imutils
import pickle
import cv2
import os


# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--dataset", required=True,
      help="path to input directory of faces + images")
ap.add_argument("-e", "--embeddings", required=True,
      help="path to output serialized db of facial embeddings")
ap.add_argument("-d", "--detector", required=True,
      help="path to OpenCV's deep learning face detector")
ap.add_argument("-m", "--embedding-model", required=True,
      help="path to OpenCV's deep learning face embedding model")
ap.add_argument("-c", "--confidence", type=float, default=0.5,
      help="minimum probability to filter weak detections")
```

```
args = vars(ap.parse_args())

# load our serialized face detector from disk
print("[INFO] loading face detector...")
protoPath = os.path.sep.join([args["detector"], "deploy.prototxt"])
modelPath = os.path.sep.join([args["detector"],
      "res10_300x300_ssd_iter_140000.caffemodel"])
detector = cv2.dnn.readNetFromCaffe(protoPath, modelPath)

# load our serialized face embedding model from disk
print("[INFO] loading face recognizer...")
embedder  =  cv2.dnn.readNetFromTorch(args["embedding_model"])

# grab the paths to the input images in our dataset
print("[INFO] quantifying faces...")
imagePaths = list(paths.list_images(args["dataset"]))

# initialize our lists of extracted facial embeddings and
# corresponding people names
knownEmbeddings = []
knownNames = []

# initialize the total number of faces processed
total = 0

# loop over the image paths
for (i, imagePath) in enumerate(imagePaths):
      # extract the person name from the image path
      print("[INFO] processing image {}/{}".format(i + 1,
            len(imagePaths)))
      name = imagePath.split(os.path.sep)[-2]
```

```
# load the image, resize it to have a width of 600 pixels (while
# maintaining the aspect ratio), and then grab the image
# dimensions
image = cv2.imread(imagePath)
image = imutils.resize(image, width=600)
(h, w) = image.shape[:2]

# construct a blob from the image
imageBlob = cv2.dnn.blobFromImage(
	cv2.resize(image, (300, 300)), 1.0, (300, 300),
	(104.0, 177.0, 123.0), swapRB=False, crop=False)

# apply OpenCV's deep learning-based face detector to localize
# faces in the input image
detector.setInput(imageBlob)
detections = detector.forward()

# ensure at least one face was found
if len(detections) > 0:
	# we're making the assumption that each image has only ONE
	# face, so find the bounding box with the largest probability
	i = np.argmax(detections[0, 0, :, 2])
	confidence = detections[0, 0, i, 2]

	# ensure that the detection with the largest probability also
	# means our minimum probability test (thus helping filter out
	# weak detections)
	if confidence > args["confidence"]:
		# compute the (x, y)-coordinates of the bounding box for
		# the face
		box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
		(startX, startY, endX, endY) = box.astype("int")
```

```
                # extract the face ROI and grab the ROI dimensions
                face = image[startY:endY, startX:endX]
                (fH, fW) = face.shape[:2]

                # ensure the face width and height are sufficiently
large
                if fW < 20 or fH < 20:
                      continue

                # construct a blob for the face ROI, then pass the blob
                # through our face embedding model to obtain the 128-d
                # quantification of the face
                faceBlob = cv2.dnn.blobFromImage(face, 1.0 / 255,
                      (96, 96), (0, 0, 0), swapRB=True, crop=False)
                embedder.setInput(faceBlob)
                vec = embedder.forward()

                # add the name of the person + corresponding face
                # embedding to their respective lists
                knownNames.append(name)
                knownEmbeddings.append(vec.flatten())
                total += 1

# dump the facial embeddings + names to disk
print("[INFO] serializing {} encodings...".format(total))
data = {"embeddings": knownEmbeddings, "names": knownNames}
f = open(args["embeddings"], "wb")
f.write(pickle.dumps(data))
f.close()
```

train_model.py

```
# USAGE
# python train_model.py --embeddings output/embeddings.pickle   --
recognizer output/recognizer.pickle --le output/le.pickle

# import the necessary packages
from sklearn.preprocessing import LabelEncoder
from sklearn.svm import SVC
import argparse
import pickle

# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-e", "--embeddings", required=True,
      help="path to serialized db of facial embeddings")
ap.add_argument("-r", "--recognizer", required=True,
      help="path to output model trained to recognize faces")
ap.add_argument("-l", "--le", required=True,
      help="path to output label encoder")
args = vars(ap.parse_args())

# load the face embeddings
print("[INFO] loading face embeddings...")
data = pickle.loads(open(args["embeddings"], "rb").read())

# encode the labels
print("[INFO] encoding labels...")
le = LabelEncoder()
labels = le.fit_transform(data["names"])
```

```
# train the model used to accept the 128-d embeddings of the face and
# then produce the actual face recognition
print("[INFO] training model...")
recognizer = SVC(C=1.0, kernel="linear", probability=True)
recognizer.fit(data["embeddings"], labels)

# write the actual face recognition model to disk
f = open(args["recognizer"], "wb")
f.write(pickle.dumps(recognizer))
f.close()

# write the label encoder to disk
f = open(args["le"], "wb")
f.write(pickle.dumps(le))
f.close()
```

recognize_video_3.py

```
# USAGE
# python recognize_video_3.py --detector face_detection_model --embedding-
model openface_nn4.small2.v1.t7 --recognizer output/recognizer.pickle --le
output/le.pickle

# import the necessary packages
from  urllib.request import urlopen
from imutils.video import VideoStream
from imutils.video import FPS
import numpy as np
import argparse
import imutils
import serial
```

```
import pickle
import time
import cv2
import os

# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-d", "--detector", required=True,
      help="path to OpenCV's deep learning face detector")
ap.add_argument("-m", "--embedding-model", required=True,
      help="path to OpenCV's deep learning face embedding model")
ap.add_argument("-r", "--recognizer", required=True,
      help="path to model trained to recognize faces")
ap.add_argument("-l", "--le", required=True,
      help="path to label encoder")
ap.add_argument("-c", "--confidence", type=float, default=0.5,
      help="minimum probability to filter weak detections")
args = vars(ap.parse_args())

# load our serialized face detector from disk
print("[INFO] loading face detector...")
protoPath = os.path.sep.join([args["detector"], "deploy.prototxt"])
modelPath = os.path.sep.join([args["detector"],
      "res10_300x300_ssd_iter_140000.caffemodel"])
detector = cv2.dnn.readNetFromCaffe(protoPath, modelPath)

#serial monitor obj
ser= serial.Serial('COM5',9600,timeout=1)
ser.write(b'L')

# load our serialized face embedding model from disk
print("[INFO] loading face recognizer...")
```

```
embedder  =  cv2.dnn.readNetFromTorch(args["embedding_model"])

# load the actual face recognition model along with the label encoder
recognizer = pickle.loads(open(args["recognizer"], "rb").read())
le = pickle.loads(open(args["le"], "rb").read())

# initialize the video stream, then allow the camera sensor to warm up
print("[INFO] starting video stream...")
#url = "http://192.168.1.2:8080/shot.jpg"
vs = VideoStream(src=0).start()
time.sleep(2.0)

# start the FPS throughput estimator
fps = FPS().start()

# loop over frames from the video file stream
while True:
      # grab the frame from the threaded video streaur
      frame = vs.read()
      key = cv2.waitKey(1) & 0xFF
      if key == ord("q"):
            ser.write(b'L')
            break

      msg = ser.readline()
      if msg == b'open\r\n':
                continue
      #imgResp = urlopen(url)
      #imgNp  =  np.array(bytearray(imgResp.read()),dtype=np.uint8)
      #frame = cv2.imdecode(imgNp,-1)

      # resize the frame to have a width of 600 pixels (while
```

```
# maintaining the aspect ratio), and then grab the image
# dimensions
frame = imutils.resize(frame, width=600)
(h, w) = frame.shape[:2]

# construct a blob from the image
imageBlob = cv2.dnn.blobFromImage(
	cv2.resize(frame, (300, 300)), 1.0, (300, 300),
	(104.0, 177.0, 123.0), swapRB=False, crop=False)

# apply OpenCV's deep learning-based face detector to localize
# faces in the input image
detector.setInput(imageBlob)
detections = detector.forward()

# loop over the detections
for i in range(0, detections.shape[2]):
	# extract the confidence (i.e., probability) associated with
	# the prediction
	confidence = detections[0, 0, i, 2]

	# filter out weak detections
	if confidence > args["confidence"]:
		# compute the (x, y)-coordinates of the bounding box for
		# the face
		box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
		(startX, startY, endX, endY) = box.astype("int")

		# extract the face ROI
		face = frame[startY:endY, startX:endX]
		(fH, fW) = face.shape[:2]
```

```
                # ensure the face width and height are sufficiently
                large
                if fW < 20 or fH < 20:
                      continue

                # construct a blob for the face ROI, then pass the blob
                # through our face embedding model to obtain the 128-d
                # quantification of the face
                faceBlob = cv2.dnn.blobFromImage(face, 1.0 / 255,
                      (96, 96), (0, 0, 0), swapRB=True, crop=False)
                embedder.setInput(faceBlob)
                vec = embedder.forward()

                # perform classification to recognize the face
                preds = recognizer.predict_proba(vec)[0]
                j = np.argmax(preds)
                proba =  preds[j]
                name = le.classes_[j]

                # draw the bounding box of the face along with the
                # associated probability
                text = "{}: {:.2f}%".format(name, proba * 100)
                y = startY - 10 if startY - 10 > 10 else startY + 10
                cv2.rectangle(frame, (startX, startY), (endX, endY),
                      (0, 0, 255), 2)
                cv2.putText(frame, text, (startX, y),
                      cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 0, 255), 2)
                #sending output to arduino
                if not name == "unknown":
                     ser.write(b'H')

# update the FPS counter
```

```
		fps.update()

		# show the output frame
		cv2.imshow("Frame", frame)
		key = cv2.waitKey(1) & 0xFF

		# if the `q` key was pressed, break from the loop
		if key == ord("q"):
			ser.write(b'L')
			break

# stop the timer and display FPS information
fps.stop()
print("[INFO] elasped time: {:.2f}".format(fps.elapsed()))
print("[INFO] approx. FPS: {:.2f}".format(fps.fps()))

# do a bit of cleanup
cv2.destroyAllWindows()
vs.stop()
```

CHAPTER-6

RESULTS

Prototype of the safe works as it was expected and successfully detects the owners face with the implementation of software code and opens the mechanical latch, designed with precision to increase the security of the system and closes only when prompted by the owner.

The motive of the project is achieved by this simple prototype with the objective of improving the security and safeguard the properties of the owner from theft and burglary.

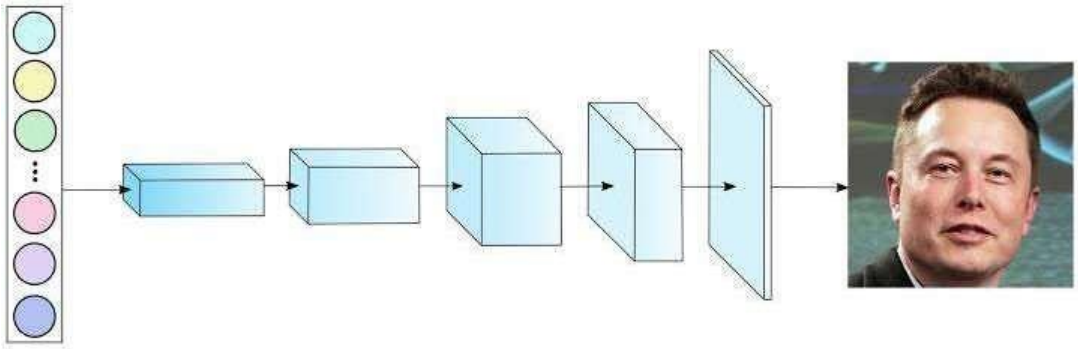


FIGURE 6.1

An embedding is the collective name for mapping input features to vectors.

In a facial recognition system, these inputs are images containing a subject's face, mapped to a numerical vector representation.

These embeddings can be used as feature inputs into classification, clustering, or regression task

Since these vectors are represented in shared vector space, vector distance can be used to calculate the similarity between two vectors.

In a facial recognition context, this can vector distance applied to calculate how similar two faces are.

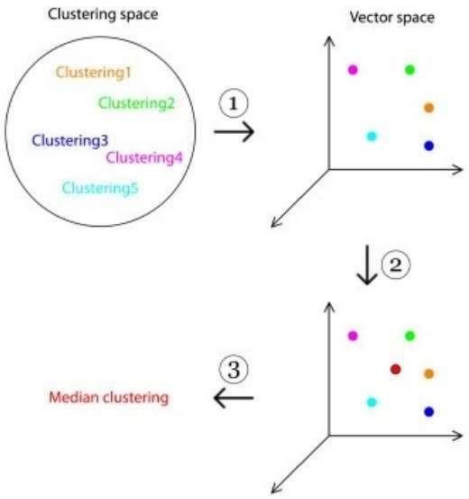


FIGURE 6.2

Additionally, these embeddings can be used as feature inputs into classification, clustering, or regression task.

The model also uses a triplet loss function to make more accurate predictions of the trained face.

In triplet loss function a base image, a postive image and a negative image is given.

The distance between the base image and positive image is reduced and the distance between the negative image and base image is increased.

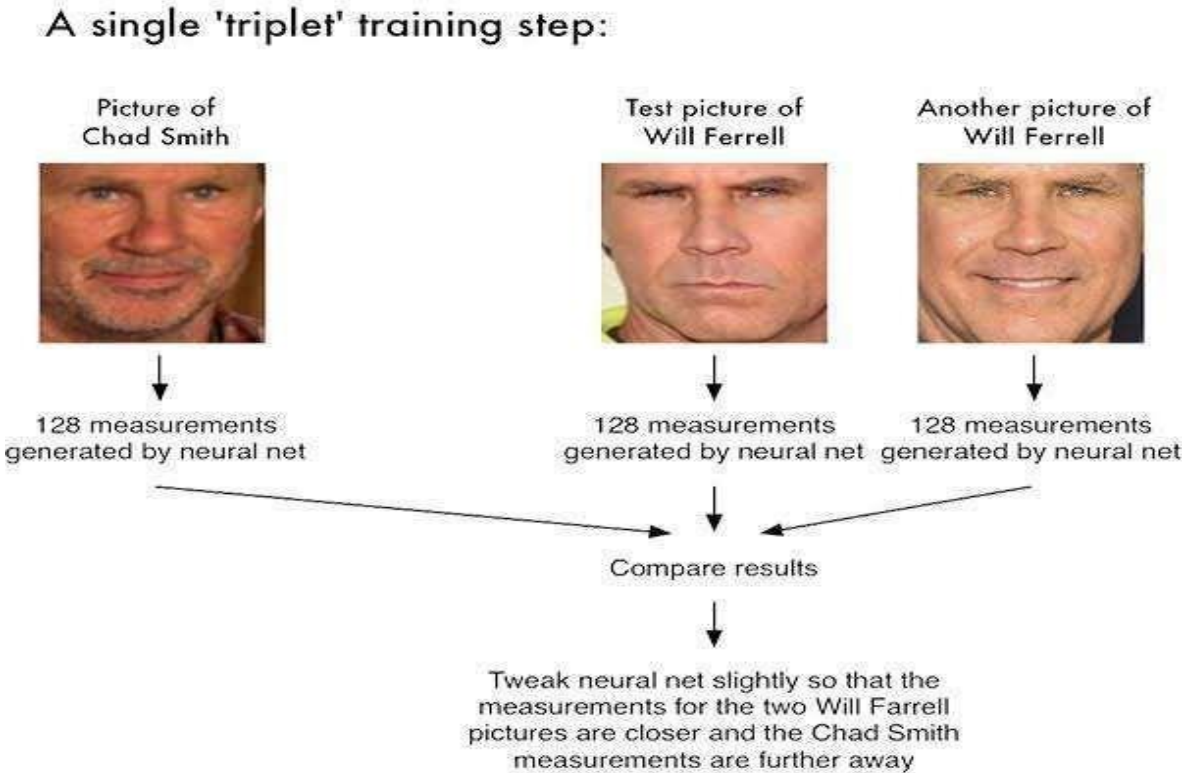

A single 'triplet' training step:
Picture of Chad Smith
Test picture of Will Ferrell
Another picture of Will Ferrell
128 measurements generated by neural net
128 measurements generated by neural net
128 measurements generated by neural net
Compare results
Tweak neural net slightly so that the measurements for the two Will Farrell pictures are closer and the Chad Smith measurements are further away

FIGURE 6.3

CHAPTER-8

CONCLUSION

We have designed and fabricated the prototype of our project (Biometric Face Recognition Security System) and we are here to conclude that the methodology used in this project helps to the further development of the existing security systems in various fields.

Though we made the design as a prototype, in our view it strengthens the disadvantages in the ancient methodology and converted to an ergonomical product as it does not normally involve any physical locking mechanism rather the whole process is happening in the central computer system which no one other than the owner has access to.

The application of IOT which enables the owner to control the system over the Internet can be implemented to access the camera of the security system from any side of the world.

By incorporating the latest solenoid enabled lock other than the mechanical rack and pinion can ensure the system works at any situation, by reducing the error of servos system.

Using Raspberry Pi can help in faster processing of the face and increase the speed of detection and recognition of the face.

Instead of using 64 points 128 embeddings of the face, we can implement the use of 208 points embeddings which captures the image at 3D format and quantifying them enables the system to identify the difference between an image and a real person.

CHAPTER-9

REFERENCES

1. American National Standards Institute, Gear Nomenclature, Definitions of Terms with Symbols (ANSI/AGMA 1012-G05 ed.), American Gear Manufacturers Association

2. M. Turk and A. Pentland. Eigenfaces for recognition.

3. L. Wiskott, J.-M. Fellous, N. Krüger and C.v.d. Malsburg. Face recognition by elastic bunch graph matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19:775-779, 1997

4. G. Heusch, Y. Rodriguez, and S. Marcel. Local Binary Patterns as an Image Preprocessing for Face Authentication. In IEEE International Conference on Automatic Face and Gesture Recognition (AFGR), 2006.

5. Bishop, C. M. (2006) Pattern Recognition and Machine Learning. Chapter 5: Neural Networks.
   Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview: Neural Networks 61: 85-117.
   Bengio, Y., LeCun, Y., Hinton, G. (2015). Deep Learning. Nature 521: 436-44.
   Goodfellow, I., Bengio, Y. and Courville, A. (2016) Deep Learning. MIT Press.

6. Bingo, Y.; Courville, A.; Vincent, P. (2013). "Representation Learning: A Review and New Perspectives". IEEE Transactions on Pattern Analysis and Machine Intelligence. 35 (8): 1798–1828

7. Schmidhuber, J. (2015). "Deep Learning in Neural Networks: An Overview". Neural Networks