**A Project Report on**

**LINKING SMART CAMERAS WITH ALARM SYSTEM BY USING CONVOLUTIONAL NEURAL NETWORKS**

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

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**BY**

**R. LOHITH VENKATA VEERENDRANADH 188X1A0537**

**V. URMILA SAI 188X1A0553**

**T. SRI NIKESH REDDY 188X1A0543**

**CH. PREM CHAND 188X1A0512**

**Under the Esteemed Guidance**

**Mrs. P. Naga Lakshmi**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY**

**Approved by AICTE- New Delhi, Accredited by NAAC A Grade and NBA Accredited**

**Permanently Affiliated to Jawaharlal Nehru Technological University, Kakinada**

**NH-5, Chowdavaram, Guntur**

**MAY - 2022**

**KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

****

###### **CERTIFICATE**

This is to certify that the project report entitled “**LINKING SMART CAMERAS WITH ALARM SYSTEM BY USING CONVOLUTIONAL NEURAL NETWORKS”** being submitted by

**R. LOHITH VENKATA VEERENDRANADH 188X1A0537**

**V. URMILA SAI 188X1A0553**

**T. SRI NIKESH REDDY 188X1A0543**

**CH. PREM CHAND 188X1A0512**

in partial fulfilment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering to the Jawaharlal Nehru Technological University, Kakinada is a record of bonafide work carried out under my guidance and supervision.

**NAME OF THE GUIDE HEAD OF THE DEPARTMENT**

|  |  |
| --- | --- |
| **Mrs. P. Naga Lakshmi**  **B. Tech, M. Tech, [Ph.D.]**  **Assistant Professor, CSE, KHIT** | **Dr.K.Venkata Subba Reddy**  **B. Tech, M. Tech, Ph.D.**  **Professor & HOD** |

**External Examiner**

**DECLARATION**

I hereby declare that the dissertation entitled “**LINKING SMART CAMERAS WITH ALARM SYSTEM BY USING CONVOLUTIONAL NEURAL NETWORKS”** submitted for the B. Tech Degree is my original work and the dissertation has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

Place: Guntur **R. LOHITH VENKATA VEERENDRANADH 188X1A0537**

**V. URMILA SAI 188X1A0553**

**T. SRI NIKESH REDDY 188X1A0543**

**CH. PREM CHAND 188X1A0512**

Date:

**ACKNOWLEDGMENT**

We profoundly grateful to express our deep sense of gratitude and respect towards our honorable chairman, our Grand Father **Sri KALLAM HARANADHA REDDY**, Chairman of Kallam group for his precious support in the college.

We are thankful to **Dr. M. UMA SANKAR REDDY**, Director, KHIT, GUNTUR for his encouragement and support for the completion of the project.

We are much thankful to **Dr. B. SIVA BASIVI REDDY,** Principal, KHIT, GUNTUR for his support during and until the completion of the project.

We are greatly indebted to **Dr. K. VENKATA SUBBA REDDY,** Professor, & Head of the department, Computer Science and Engineering, KHIT, GUNTUR for providing the laboratory facilities fully as and when required and for giving us the opportunity to carry the project work in the college.

We are also thankful to our Project Coordinators **Dr. G. SRINIVASA RAO**

and **Dr. P. RATNA BABU** who helped us in each step of our Project.

We extend our deep sense of gratitude to our Internal Guide **Mrs. P. Naga Lakshmi,**

Assistant Professor, & Head of the department and other Faculty Members & Support staff for their valuable suggestions, guidance and constructive ideas in each and every step, which was indeed of great help towards the successful completion of our project.

Place: Guntur **R. LOHITH VENKATA VEERENDRANADH 188X1A0537**

**V. URMILA SAI 188X1A0553**

**T. SRI NIKESH REDDY 188X1A0543**

**CH. PREM CHAND 188X1A0512**

Date:

**ABSTRACT**

Computational vision-based Person detection has drawn significant attention in the past decade with camera surveillance systems becoming ubiquitous. Signal and image processing methods are developed for the detection of person in open and large spaces with a range of up to 30m to the camera in visible-range (IR) video. By using Artificial Intelligence, through face detection, we will store a dataset with faces of family and relatives. If it is unauthorized person then we will set an alert and also send an email notification or message to the owner or apartment watchman. Previously, SVM algorithm is used in the existing system. Now in the proposed work Convolutional Neural network and along with SVM. Convolutional Neural Network (CNN/ConvNet) is a class of deep neural networks, most commonly applied to analyze visual imagery.

**TABLE OF CONTENTS**

**TITLE PAGE NO.**

**Chapter 1 Introduction 1**

1.1 Purpose 1 – 2

1.2 Problem Statement 2

1.3 Scope 2

**Chapter 2 Literature Survey 3**

2.1 Literature survey on bank locker security system 3

2.2 Literature survey on door lock security systems 3

2.3 A review of intelligent lock system 3

2.4 A review on secured bank locker system using fingerprint, image and

rfid technique 3

**Chapter 3 System Analysis & Feasibility Study** 4 - 5

3.1 Existing System 4

3.1.1 Disadvantage of Existing System 4

3.2 Proposed System 4

3.2.1 Advantage of Proposed System 4

3.3. Feasibility Study 5

3.3.1 Economically Feasibility 5

3.3.2 Technical Feasibility 5

3.3.3 Social Feasibility 5

**Chapter 4 System Requirements 6 -7**

4.1 Software Requirements 6

4.2 Hardware Requirements 6

4.3 Functional Requirements 6

4.4 Non-Functional Requirements 7

**Chapter 5 System Design 8 - 13**

5.1 Data Flow Diagram 8

5.2 UML Diagrams 9

5.2.1 Use Case Diagram 10

5.2.2 Class Diagram 11

5.5.3 Sequence Diagram 12

5.5.4 Activity Diagram 13

**Chapter 6 Implementation 14 - 25**

6.1 Module Description 14

6.2 Algorithm 14

6.2.1 What is Deep Learning 14-15

6.2.2 How Deep Learning works 15

6.2.3 SVM (Support Vector Machine) 15

6.2.4 CNN (Convolutional Neural Networks) 16 – 18

6.2.5 Algorithm 18

6.3 Coding (Source Code) 19 - 22

6.4 Packages Imported 23

6.5 Python 23

6.6 Modules used in project 23 - 25

6.6.1 NumPy 23

6.6.2 Pickle 24

6.6.3 Cv2 24

6.6.4 PlaySound 24

6.6.5 TeleBot 24

6.6.6 Time 25

**Chapter 7 System Testing 26 - 27**

7.1 Testing Strategies 26

7.1.1 Unit Testing 26

7.1.2 Integration Testing 26

7.1.3 User Interface Testing 26

7.2 Test Cases 27

**Chapter 8 Output Screens 28 - 30**

**Chapter 9 Conclusion 31 - 32**

**Chapter 10 References 33 - 34**

**LIST OF FIGURES**

NAME PAGE NO:

**5.1 Data Flow Diagram**

5.1 Data Flow 8

**5.2 UML Diagrams**

5.2.1 Use Case Diagram 10

5.2.2 Class Diagram 11

5.2.3 Sequence Diagram 12

5.2.4 Activity Diagram 13

**6.2 Algorithms**

6.2.4.1 Feature Detector 16

6.2.4.2 Convolutional Layer 17

6.2.4.3 Pixel Separation 17

**8 Output Screens**

8.1 Authorized Person Detection 28

8.2 Authorized Person with Matching Percentage 29

8.3 Unauthorized Person Detection 29

8.4Unauthorized Person with Matching Percentage 30

8.5Unauthorized Person Detection of two Persons 30

**LIST OF TABLES**

**NAME PAGE NO**

7.2 Test Cases

7.2 Test Cases 27

**CHAPTER- 1**

**INTRODUCTION**

**1.1 Purpose:**

Over the last decade humans are solving very complex and sophisticated real- world problems. But we humans are lag in the controlling unauthorized access. In case of digital solutions, we are eradicating these unknown entries by using password encryptions. But we humans are very lag in the areas where we can’t prevent the unauthorized access. Now Deep Learning have capabilities to do these types of tasks included almost all the real-world domains such as healthcare, autonomous vehicle (AV), business applications, natural language processing (NLP), intelligent robots, gaming, climate modeling, voice, and image processing. These DL algorithm’s learning is typically based on trial-and-error method quite opposite of conventional algorithms, which follows the programming instructions based on decision statements like if-else. In particular, the study is focused on prevention for the unauthorized persons to enter into our premises. This study aims to provide an early prevention harm from the harmful or unauthorized persons by simply alarming us by using a buzzer and also send a message to the owner. This project targets to develop identification device in order to handle these unauthorized people to access or enter our premises. And our device is termed as user friendly because it doesn’t have much more complexities.

Face Detection is a computer technology being used in a variety of application that identifies human faces in digital images. Face detection also refers to the psychological process by which humans locate and attend to faces in a visual scene. Face detection can be regarded as a specific case of object-class detection. In object-class detection, the task is to find the locations and sizes of all objects in an image that belong to given class. Examples include upper torses, pedestrians and cars. Face detection algorithms focus on the detection of frontal human faces. It is analogous to image detection in which the image of a person is matched bit by bit. Image matches with the image stores in database. Any facial feature changes in the database will invalidate the matching process.

A Facial recognition system is a technology capable of matching a human face from a digital image or a video frame against a database of faces, typically employed to authenticate users through ID verification services, works by pinpointing and measuring facial features from a given image.

When initially a form of computer application, facial recognition systems have been seen wider uses in recent times on smartphones and in other forms of technology, such as robotics. Because computerized facial recognition involves the measurement of a human’s physiological characteristics facial recognition systems are categorized as biometrics. Although the accuracy of facial recognition systems as a biometric technology is lower than iris recognition and fingerprint recognition, it is widely adopted due to its contactless process. Facial recognition systems have been deployed in advanced human-computer interaction, video surveillance

and automatic indexing images.

Proposed system is used to provide security in the society. The main purpose is to reduce the robbing and crimes in the society. We can place a sensor in CCTV Camera that uses Artificial Intelligence to detect the faces by storing large amount of data of family and relatives. After dataset is created, if any unknown person comes then an alert mail and message sent to owner.

**1.2 Problem Statement:**

Linking Smart cameras with alarm systems to avoid triggering a false alarm by birds or animals or vibrations due to wind, loud sounds, etc.

**1.3 Scope:**

In this project, our aim is to identify the unknown persons enter into our environments. Here we will recognize those people using Deep Learning.

**CHAPTER- 2**

**LITERATURE SURVEY**

2.1 **Title: LITERATURE SURVEY ON BANK LOCKER SECURITY SYSTEM**

**Authors:** M . Sabareesh , Sujitha Kannadasan, R.Swathi

**Description:**The main aim of this paper is to provide the high level security of bank lockers by using robotics. As we have a major threat over the security we depend upon the bank to guard our inestimable properties.

link : <https://www.ijsr.in/upload/436009143Chapter_33.pdf>

2.2 **Title:** **LITERATURE SURVEY ON DOOR LOCK SECURITY SYSTEMS**

**Authors:** PradnyaR. Nehete, J. P.Chaudhari, S. R. Pachpande, K. P. Rane, PhD

**Description:** Today people are facing more problems about security in all over world.Here in this paper, trying to reproduce the comprehensive literature study related to the various door locks and gate security systems that are necessary in the fields such as home, industries and vehicle security where possibilities of incursion are increasing day by day.

Link:http://www.makeitortakeit.in/documents/70/56.\_Android\_Based\_Industrial\_security\_locking\_system.pdf

2.3 **Title: A REVIEW OF INTELLIGENT LOCK SYSTEM**

**Authors:**S. O. Anaza , J. D. Jiya2 and Y. S.Haruna

**Description:** Existing literature does not present first hand information to researcher to ascertain the research gap. This paper presents a review of some literatures in intelligent security lock systems, by presenting their concepts, the advantages and the drawback of such concepts and the possible modification.

Link:http://www.ajer.org/papers/v6(06)/B06060915.pdf

2.4 **Title:** **A REVIEW ON SECURED BANK LOCKER SYSTEM USING FINGERPRINT, IMAGE AND RFID TECHNIQUE**

**Authors:** Rekha Dewangan, Vishnu Kumar Mishra, Megha Mishra

**Description:**In this system only authentic people can recover money from a locker. We have implemented a locker security system based on RFID, finger print and Image processing technology containing a door locking system which can activate, authenticate, and validate the user and unlock the door in real time for locker secure access.

Link:https://www.ijarcce.com/upload/2016/february-16/IJARCCE%2058.pdf

**CHAPTER- 3**

**SYSTEM ANALYSIS AND FEASIBILITY STUDY**

**3.1Existing System:**

The recent versions of home alarm systems mostly include wireless surveillance cameras. Another interesting feature of the recent home alarm system is the mobile application that allows users to interact with the system through their phone, which allow the user to activate the alarm, monitor through the surveillance camera and check the sensor’s status and others, via the mobile application. SVM algorithm is used in the existing system. it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate.

**3.1.1 Disadvantages:**

* In SVM classifier, Learning takes long time. So automatically this impacted on execution speed also. i.e., low execution speed.
* It takes long time for image comparison i.e., low flexible.
* Memory Requirements are high.
  1. **Proposed System:**

For CCTV cameras, we use the normal laptop cameras. If the stranger or unauthorized object comes Infront of camera the alarm will be activated. Major use of proposed work is providing safety, by using of Convolutional Neural Network. CNN is a powerful algorithm for image processing. These algorithms are currently the best algorithms we have for the automated processing of images. Many companies use these algorithms to do things like identifying the objects in an image. CNN can automatically detect the important features without any human supervision. For example, given many pictures of cats and dogs it learns distinctive features for each class by itself. CNN is also computationally efficient. Convolutional Neural Networks specialized for applications in image & video recognition. CNN is mainly used in image analysis tasks like Image recognition, Object detection & Segmentation.

**3.2.1Advantages:**

* In CNN classifier, Learning takes short time i.e., High execution speed.
* Comparison of images are fast i.e., high flexible.
* It has low memory requirements.

**3.3 FEASIBILITY STUDY**

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**3.3.1 ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### 3.3.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**3.3.3 SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**CHAPTER-4**

**SYSTEM REQUIREMENTS**

**4.1 SOFTWARE REQUIREMENTS**

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

* Operating system: Windows 7.
* Coding language: Python.
* IDE: Python IDLE.

**4.2 HARDWARE REQUIREMENTS**

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

* + System: Intel Core i7 2.4 GHz.
  + Hard Disk: 500 GB.
  + RAM: 4GB.

### 4.3 FUNCTIONAL REQUIREMENTS

The Functional Requirements Specification gives the operations and activities that a system must be able to perform. Functional requirements should include functions performed by specific screens, outlines of work-flows performed by the system, and other business or compliance requirements the system must meet. It also depends upon the type of software, expected users and the type of system where the software is used. Here the model was designed by following:

* Data Set
* Data Preprocessing
* Training And Testing
* Modelling
* Predicting

**4.4 NON-FUNCTIONAL REQUIREMENTS**

In systems engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. The nonfunctional requirements can be considered as quality attributes of a system.

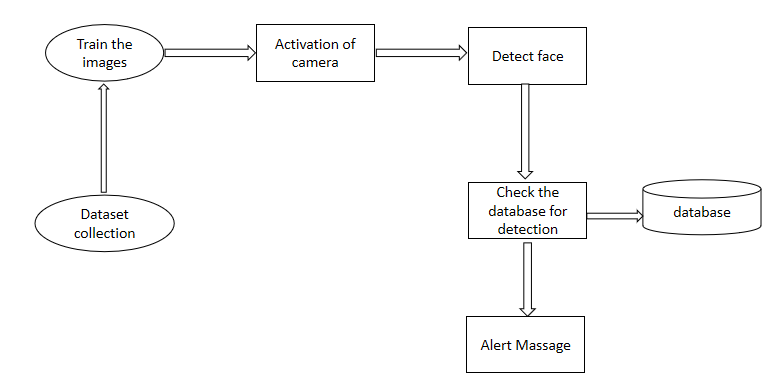
* Performance
* Reliability
* Efficiency
* Availability
* Maintainability
* Usability requirement
* Serviceability requirement
* Manageability requirement
* Recoverability requirement
* Security requirement

**CHAPTER- 5**

**SYSTEM DESIGN**

**5.1 DATA FLOW DIAGRAM:**

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

****

**Fig 5.1 Data Flow Diagram**

**5.2 UML DIAGRAMS:**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

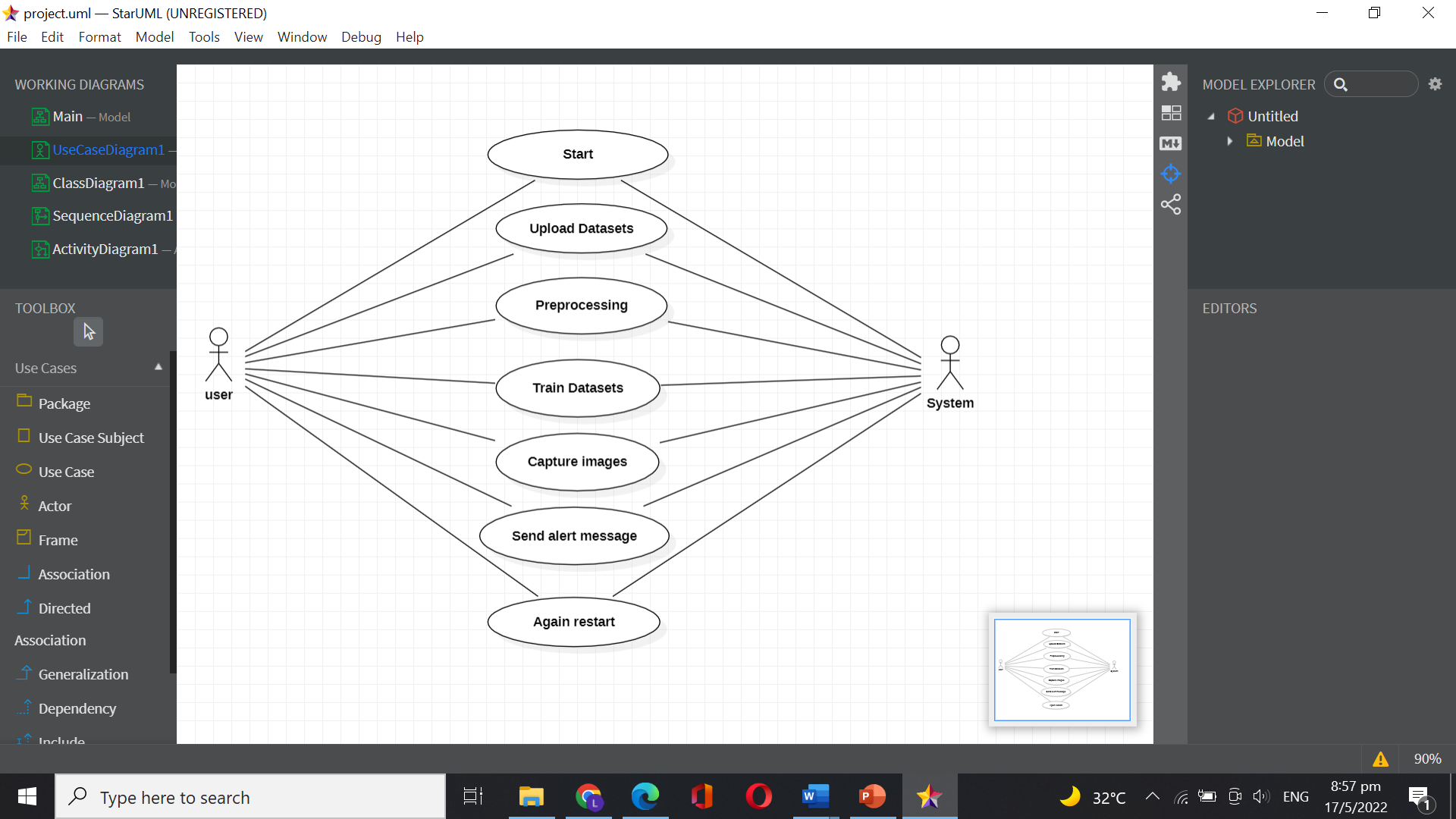
**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Integrate best practices.

**5.2.1 USE CASE DIAGRAM:**

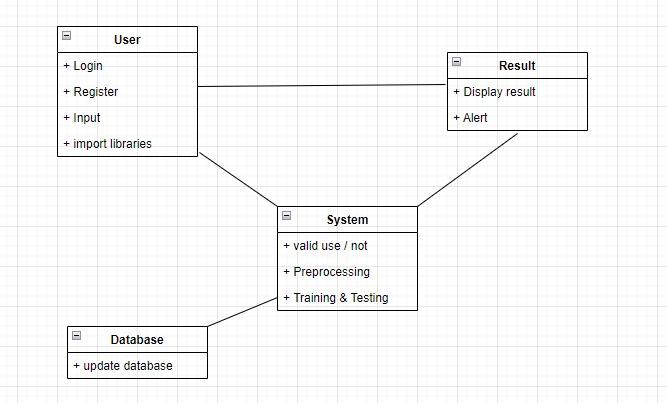
A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



**Fig 5.2.1 Use Case Diagram**

**5.2.2 CLASS DIAGRAM:**

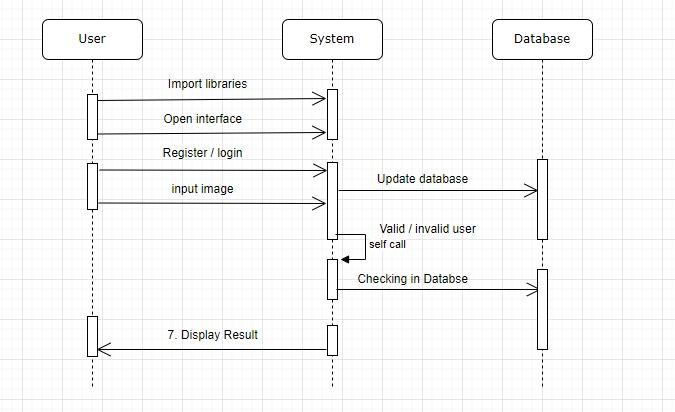
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



**Fig 5.2.2 Class Diagram**

**5.2.3 Sequence diagram:**

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".



**Fig 5.2.3 Sequence Diagram**

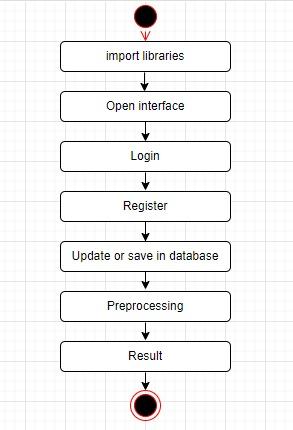
**5.2.4 Activity diagram:**

Activity diagram is essentially a fancy flowchart: Activity and state diagrams are related. State chart diagram focuses on objects undergoing a process. An activity diagram focuses on the flow of activities involved in a single process. The activity diagram shows the activities depend on one another.

An activity represents the performance of the task or duty in a workflow. It may also represent the execution of a statement in a procedure. You can share activities between state machines. However, transitions cannot be shared.

Activity diagrams provide a way to model the workflow of a business process, code specific information such as a class operation. The transitions are implicitly triggered by completion of the actions in the source activities.

The main difference between activity and state chart diagram is activities are activity centric, while state chart diagrams are state centric.

****

**Fig 5.2.4 Activity Diagram**

**CHAPTER- 6**

**IMPLEMENTATION**

**6.1 MODULE DESCRIPTION**

The modules are required for effective purposes. They are,

➢ **User:** Here the user will preprocess the data which includes detecting the null values and dropping them. The outlier values of the dataset are updated by their median values.

➢ **Dataset Training:** Here the user has to add the dataset to the training purpose. The dataset is divided into two parts for training and testing in size of 80% and 20%.

➢ **Modelling:** Here the system will train the model by installing requires libraries. Now the system will be trained with some data from the dataset.

➢ **Interface Design:** Here we will define the inputs that should be available and sufficient for the system to predict the natural gas price to the user.

➢ **Prediction of Price:** The valid input values provided by the user will be used by the specific algorithm selected by the user to predict the natural gas price of that specific date.

➢ **System Module:** The user after selecting the specific date for which he wants to know the predicted price, the algorithm will predict the price based on the trained dataset.

**6.2 Algorithms:**

## 6.2.1 What Is Deep Learning?

Deep learning is an [artificial intelligence (AI)](https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp) 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](https://www.investopedia.com/terms/m/machine-learning.asp) in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.

### KEY TAKEAWAYS

* Deep learning is an AI function that mimics the workings of the human brain in processing data for use in detecting objects, recognizing speech, translating languages, and making decisions.
* Deep learning AI is able to learn without human supervision, drawing from data that is both unstructured and unlabeled.
* Deep learning, a form of machine learning, can be used to help detect fraud or money laundering, among other functions.

## 6.2.2 How Deep Learning Works?

Deep learning has evolved hand-in-hand with the digital era, which has brought about an explosion of data in all forms and from every region of the world. This data, known simply as [big data](https://www.investopedia.com/terms/b/big-data.asp), is drawn from sources like social media, internet search engines, [e-commerce](https://www.investopedia.com/terms/e/ecommerce.asp) platforms, and online cinemas, among others. This enormous amount of data is readily accessible and can be shared through [fintech](https://www.investopedia.com/tech/worlds-top-10-fintech-companies-baba/) applications like cloud computing.

However, the data, which normally is unstructured, is so vast that it could take decades for humans to comprehend it and extract relevant information. Companies realize the incredible potential that can result from unraveling this wealth of information and are increasingly adapting to AI systems for automated support.

Deep learning unravels huge amounts of unstructured data that would normally take humans decades to understand and process.

**6.2.3 SVM (SUPER VECTOR MACHINE)**

## Introduction

Mastering [machine learning algorithms](https://courses.analyticsvidhya.com/courses/introduction-to-data-science-2?utm_source=blog&utm_medium=understandingsupportvectormachinearticle) isn’t a myth at all. Most of the beginners start by learning regression. It is simple to learn and use, but does that solve our purpose? Of course not! Because you can do so much more than just Regression!

Think of machine learning algorithms as an armory packed with axes, sword, blades, bow, dagger, etc. You have various tools, but you ought to learn to use them at the right time. As an analogy, think of ‘Regression’ as a sword capable of slicing and dicing data efficiently, but incapable of dealing with highly complex data. On the contrary, ‘Support Vector Machines’ is like a sharp knife – it works on smaller datasets, but on the complex ones, it can be much stronger and powerful in building machine learning models.

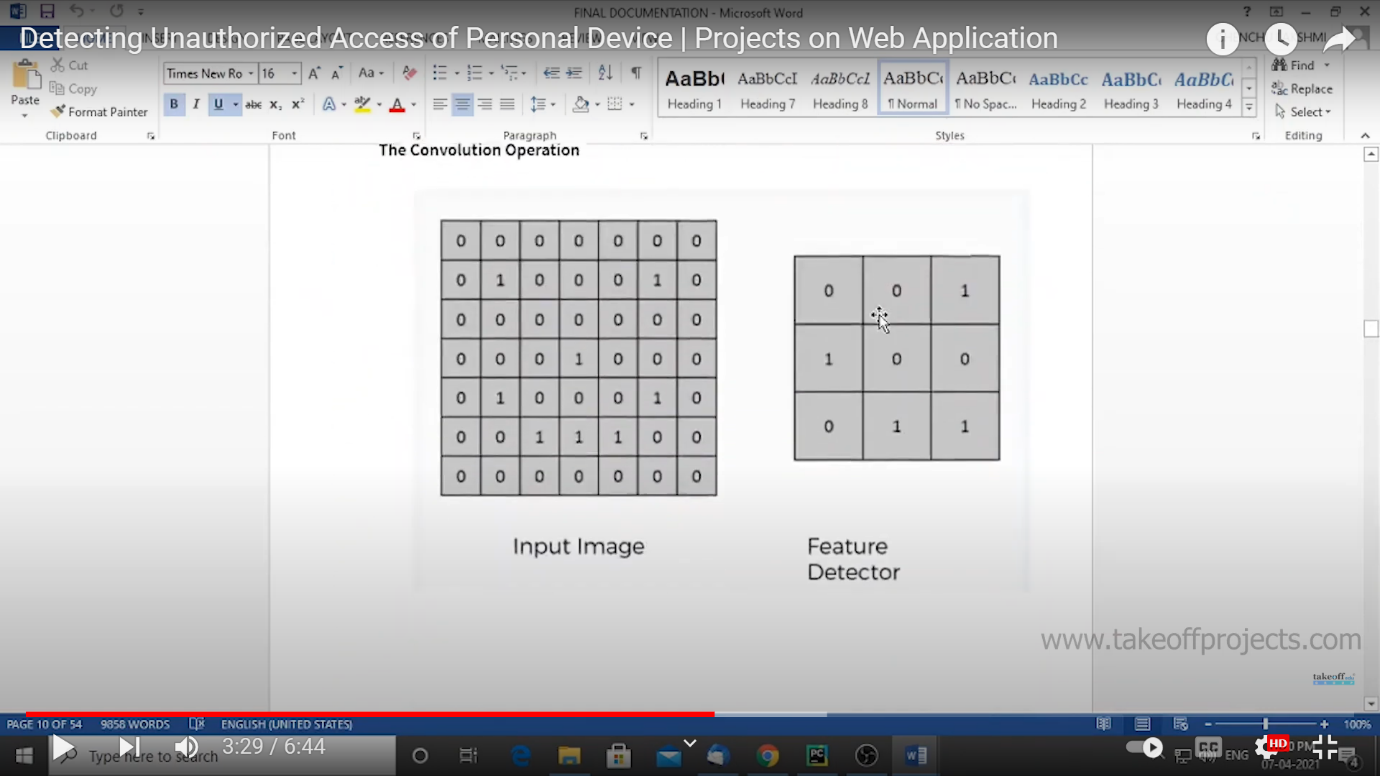
By now, I hope you’ve now mastered [Random Forest](https://www.analyticsvidhya.com/blog/2015/09/random-forest-algorithm-multiple-challenges/?utm_source=blog&utm_medium=understandingsupportvectormachinearticle), [Naive Bayes Algorithm](https://www.analyticsvidhya.com/blog/2015/09/naive-bayes-explained/?utm_source=blog&utm_medium=understandingsupportvectormachinearticle) and [Ensemble Modeling](https://courses.analyticsvidhya.com/courses/introduction-to-data-science-2?utm_source=blog&utm_medium=understandingsupportvectormachinearticle). If not, I’d suggest you take out a few minutes and read about them as well. In this article, I shall guide you through the basics to advanced knowledge of a crucial [machine learning](https://www.analyticsvidhya.com/machine-learning/?utm_source=blog&utm_medium=support-vector-machine) algorithm, [support vector machines](https://courses.analyticsvidhya.com/courses/support-vector-machine-svm-in-python-and-r?utm_source=blog&utm_medium=understaing-support-vector-machine-example-code).

**6.2.4 Convolutional Neural Networks**

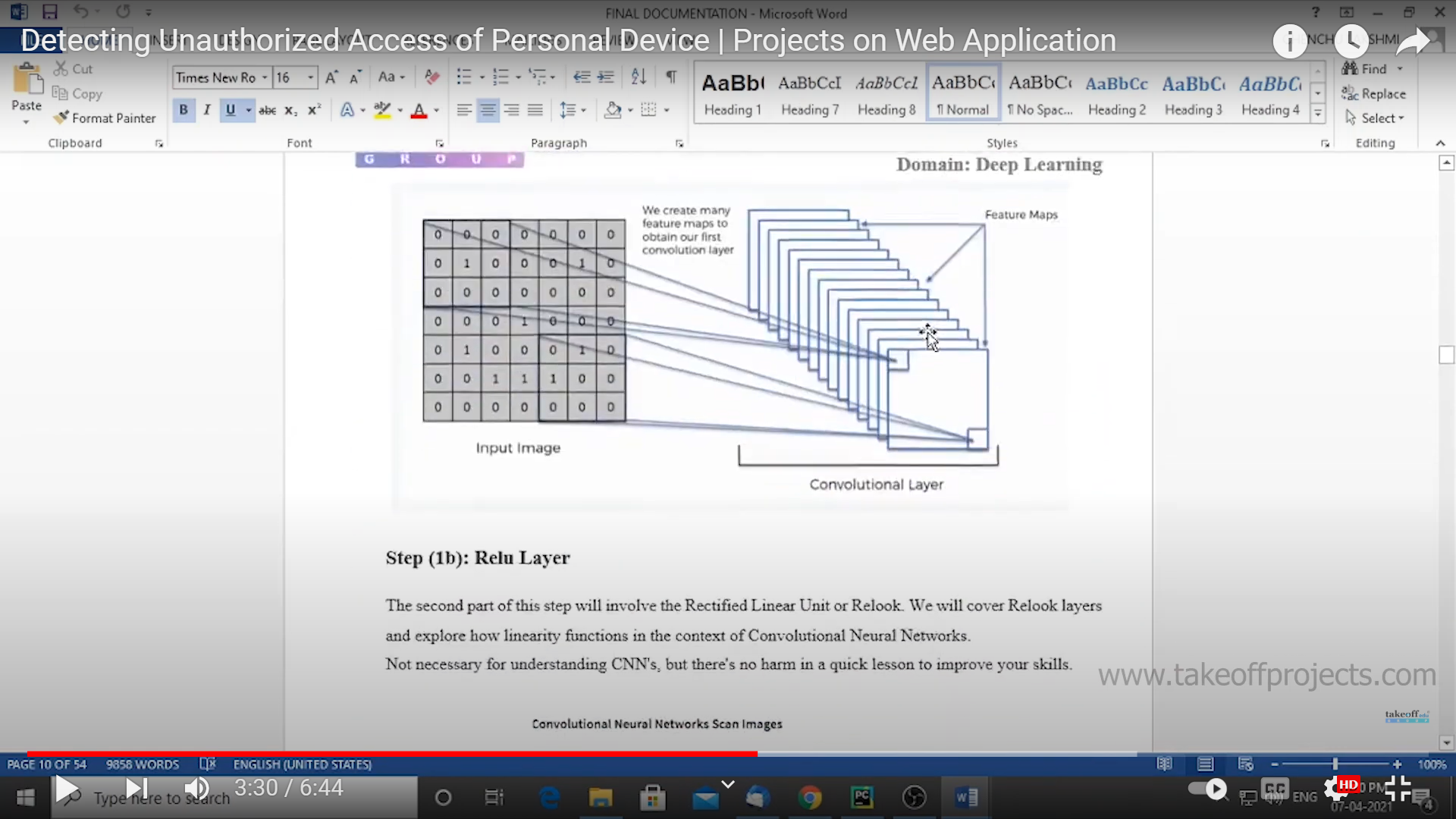
Step1: Convolutional operation

The first building block in our plan of attack is convolutional operation. In this step, we will touch on feature detectors, which basically serve as the neural network’s filters. We will also discuss feature maps, learning the parameters of such maps, how patterns are detected, the layers of detection, and how the findings are mapped out.

1.a The Convolutional Operation



**Fig..6.2.4.1 Feature Detector**

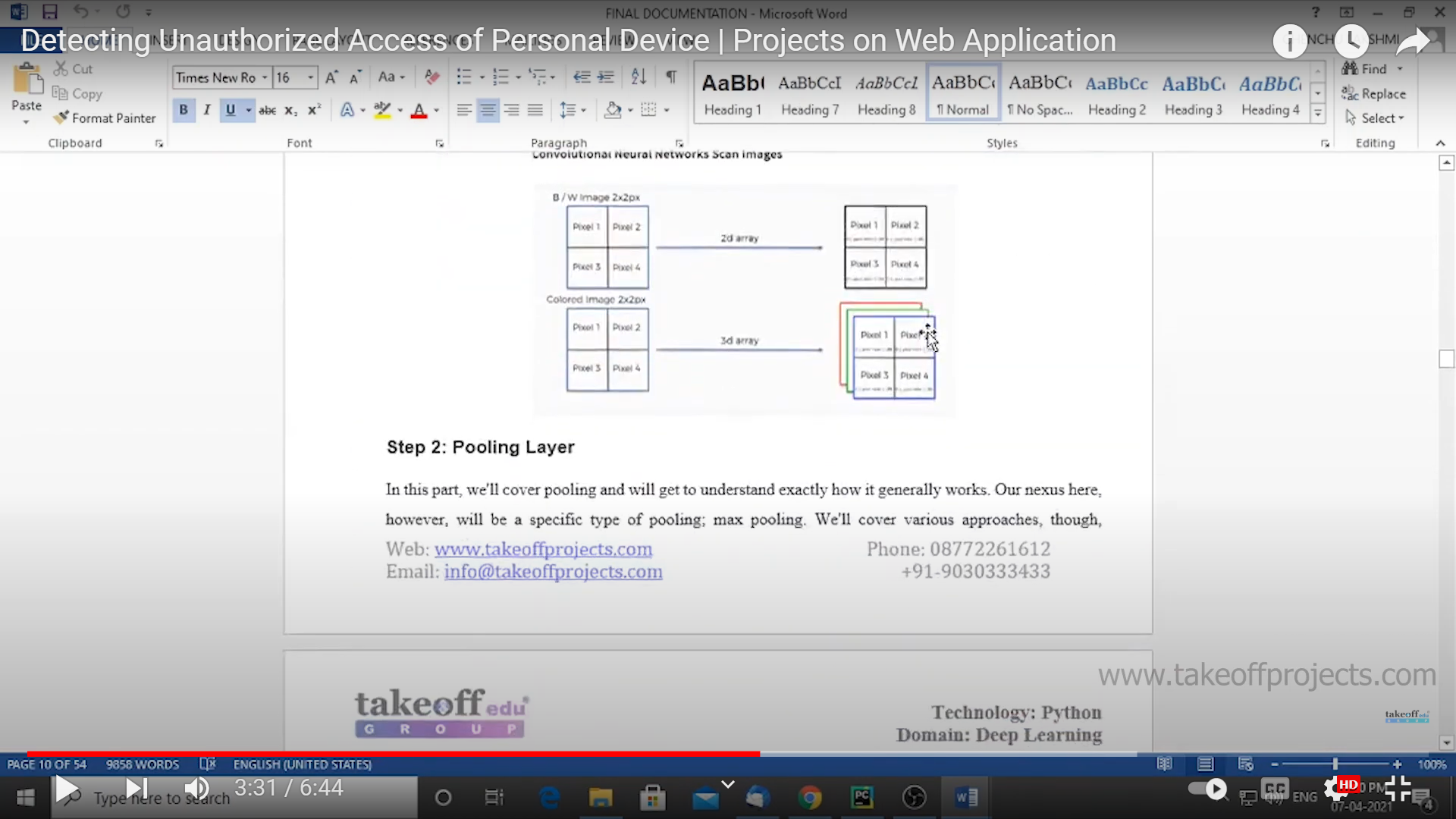


**Fig.6.2.4.2 Convolutional Layer**

Step(1b): Relu Layer

The second part of this part will involve the Rectified Linear Unit or Relook. We will cover Relook layers and explore how linearity functions in the content of convolutional Neural Networks.

Convolutional Neural Networks Scan Images



**Fig.6.2.4.3 Pixel Separation**

Step 2: Layer

In this part, we’ll cover pooling and will get to understand exactly how it generally works. Our nexus here, however, will be a specific type of pooling; max pooling. We’ll cover various approaches, though including mean (or sum) pooling. This part will end with a demonstration made using a visual interactive tool that will definitely sort the whole concept out for you.

Step 3: Flattening

This will be a brief breakdown of the flattening process and how we move from pooled to flattened layers when working with convolutional Neural Networks.

Step 4: Full Connection

In this part, everything that we covered throughout the section will be merged together. By learning this, you’ll get to envision a fuller picture of how convolutional Neural Networks operate and how the “neurons” that are finally produced learn the classification of images.

Summary

In the end, we’ll wrap everything up and give a quick recap of the concept covered in this section. If you feel like it will do you any benefit (and it probably will), you should check out the extra tutorial in which soft-ax and Cross-Entropy are covered. It’s not mandatory for the course, but you will likely come across these concepts when working with convolutional neural networks and it will do you a lot of good to be familiar with them.

**6.2.5 Algorithm:**

Step 1: Start

Step 2: Upload Datasets

Step 3: Preprocessing

Step 4: Train Datasets

Step 5: Capture Images

#By using Convolutional Neural Networks Algorithm.

Step 6: Frame Matching

#By using Support Vector Machine

Step 7: Send Alert Message

Step 8: Again Restart

* 1. **CODING (SOURCE CODE)**

import pickle

import cv2

import os

import numpy as np

import imutils

import time

from threading import Thread

import playsound

import argparse

import telebot

tb = telebot.TeleBot("5389349019:AAE52a4HnNFwfBG\_CHWTffSzcsYekAsVT1I")

ap = argparse.ArgumentParser()

ap.add\_argument("-a", "--alarm", type=str, default="abc.wav",required=False,

help="path alarm .WAV file")

args = vars(ap.parse\_args())

def sound\_alarm(path):

# play an alarm sound

playsound.playsound(path)

CONSEC\_FRAMES = 48

COUNTER = 0

ALARM\_ON = False

ALERTS\_SENT = False

BASE\_DIR = os.path.dirname(\_\_file\_\_)

print("[INFO] BASE DIR: ", BASE\_DIR)

# load our serialized face detector from disk FACE 64 LANDMARKS

print("[INFO] loading face detector...")

protoPath = os.path.join(BASE\_DIR, "face\_detection\_model/deploy.prototxt")

modelPath=os.path.join(BASE\_DIR, "face\_detection\_model/res10\_300x300\_ssd\_iter\_140000.caffemodel")

detector = cv2.dnn.readNetFromCaffe(protoPath, modelPath)

# load our serialized face embedding model from disk

embedding\_model = os.path.join(BASE\_DIR, 'nn4.small2.v1.t7')

print("[INFO] loading face recognizer...")

embedder = cv2.dnn.readNetFromTorch(embedding\_model)

# load the actual face recognition model along with the label encoder

recognizer\_file = os.path.join(BASE\_DIR, 'output/recognizer.pickle')

le\_file = os.path.join(BASE\_DIR, 'output/le.pickle')

recognizer = pickle.loads(open(recognizer\_file, "rb").read())

le = pickle.loads(open(le\_file, "rb").read())

print("[INFO] starting video stream...")

cap = cv2.VideoCapture(0)

while (True):

ret, frame = cap.read()

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 > 0.65:

# 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]

y = startY - 10 if startY - 10 > 10 else startY + 10

cv2.rectangle(frame, (startX, startY), (endX, endY),

(0, 0, 255), 2)

# draw the bounding box of the face along with the

# associated probability

if proba \* 100 >60 and name!='unknown':

text = "{}: {:.2f}%".format(name, proba \* 100)

cv2.putText(frame, text, (startX, y),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.45, (0, 0, 255), 2)

COUNTER = 0

ALARM\_ON = False

ALERTS\_SENT = False

else:

text = "{}: {:.2f}%".format('unknown', proba \* 100)

cv2.putText(frame, text, (startX, y),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.45, (0, 0, 255), 2)

COUNTER += 1

if COUNTER >= CONSEC\_FRAMES:

if not ALARM\_ON:

ALARM\_ON = True

# check to see if an alarm file was supplied,

# and if so, start a thread to have the alarm

# sound played in the background

t = Thread(target=sound\_alarm,args=(args["alarm"],))

t.deamon = True

t.start()

# draw an alarm on the frame

cv2.putText(frame, "Unkown ALERT!", (10, 30),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (0, 0, 255), 2)

if not ALERTS\_SENT:

cv2.imwrite('unknown.jpg',frame)

foto = open('unknown.jpg', 'rb')

tb.send\_photo('1131668020', foto)

print ('Unkown person detected. sending alerts....')

ALERTS\_SENT=True

cv2.imshow('Camera', frame)

key = cv2.waitKey(1) & 0xFF

if key == ord('q'):

break

cap.release()

cv2.destroyAllWindows()

* 1. **PACKAGES IMPORTED**

**PACKAGE**

* NumPy
* Pickle
* Cv2
* Playsound
* Telebot
* Time

**6.5 Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

**6.6 Modules Used in Project:**

**6.6.1 NumPy**

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

**6.6.2 Pickle**

Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it “serializes” the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script.

**6.6.3 Cv2**

cv2 is **the module import name for OpenCV-python**, "Unofficial pre-built CPU-only OpenCV packages for Python". The traditional OpenCV has many complicated steps involving building the module from scratch, which is unnecessary. I would recommend remaining with the OpenCV-python library.

**6.6.4 Playsound**

Python using some of the most popular audio libraries

**Method 1:**Using *playsound* module

Run the following command to install the packages:

pip install playsound

* The*playsound*module contains only a single function named **playsound()**.
* It requires one argument: the path to the file with the sound we have to play. It can be a *local file*, or a *URL*.
* There’s an optional second argument, **block**, which is set to *True* by default. We can set it to *False* for making the function run **asynchronously**.
* It works with both **WAV** and **MP3** files.

**6.6.5 Telebot**

The TeleBot class (defined in \_init\_.py) **encapsulates all API calls in a single class**. It provides functions such as send\_xyz (send\_message , send\_document etc.) and several ways to listen for incoming messages. Create a file called echo\_bot.py.

**Time**

## Importing time module

The time module comes with Python’s standard utility module, so there is no need to install it externally. We can simply import it using the [import statement](https://www.geeksforgeeks.org/import-module-python/).

import time

**CHAPTER- 7**

**SYSTEM TESTING**

**7.1 TESTING STRATEGIES**

**7.1.1 UNIT TESTING**

Unit testing, a testing technique using which individual modules are tested to determine if there are issues by the developer himself. it is concerned with functional correctness of the standalone modules. The main aim is to isolate each unit of the system to identify, analyze and fix the defects.

Unit Testing Techniques:

Black Box Testing - Using which the user interface, input and output are tested.

White Box Testing –Used to test each one of those function behavior is tested.

**7.1.2 INTEGRATION TESTING**

Integration Testing done upon completion of unit testing, the units or modules are to be integrated which gives raise too integration testing. The purpose of integration testing is to verify the functional, performance, and reliability between the modules that are integrated.

# 7.1.3 USER INTERFACE TESTING

User interface testing, a testing technique used to identify the presence of defects is a product/software under test by Graphical User interface [GUI].

**7.2 TEST CASES:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **INPUT** | **OUTPUT** | **RESULT** |
| **Test Case 1**  (UNIT TESTING) | The user gives the input in the form of a training with a dataset. | An output is predicted as the training is successful. | The result is that the dataset is trained. Therefore, the test case1 is passed successfully. |
| **Test Case 2**  (INTEGRATION  TESTING) | The user gives the input in the form of  Opening interface. | An output is predicted as the user opens the interface is successful. | The result is that the user opened the interface. Therefore, the testcase2 passed successfully. |
| **Test Case 3**  (USER INTERFACE TESTING) | The user gives the input in the form of  entering data. | An output is predicted as the user enters the data is successful. | The result is that the user enters the data. Therefore, the testcase3 passed successfully. |
| **Test Case 4**  (USER INTERFACE TESTING) | The user gives the input in the form of  Click on predict. | An output is predicted as the user gets a result as the price of natural gas. | The result is that the user gets the predicted price. Therefore, the testcase4 passed successfully. |

**Fig 7.2 Test Cases**

**CHAPTER- 8**

**OUTPUT SCREENS**

# User will enter specific date on which he wants to predict the price of natural gas. After submitting the data, the algorithm will predict the price of the natural gas from the trained data.

# 

# Fig. 8.1 Random Forest Algorithm Successful Prediction

# It will show the predicted price value as per the valid data given by the user in the respective fields by using trained data.

# 

# Fig. 8.2 Random Forest Algorithm Un-Successful Prediction

# In this case, the user entered the invalid data in the fields. So, the system sent displaying us an error message.

# 

# Fig. 8.3 Decision Tree Algorithm Successful Prediction

# Image is not matched (unknown persons) with the trained image, then it makes a sound through alarm system.

# 

# Fig. 8.4 Decision Tree Algorithm Un-Successful Prediction

# In this case, the user entered the invalid data in the fields. So, the system sent displaying us an error message.

**CHAPTER- 9**

**CONCLUSION**

# It has always been a difficult task to predict the exact daily price of natural gas price. Many factors such as political events, general economic conditions, and traders’ expectations may have an influence on it. But here, based on the past and present traits, we were able to achieve up to 97% accuracy in predicting the price of any given date. Albeit, it’s impossible to predict unexpected scenarios such as acts of warfare or terrorism. But, the benefits of having reliable information of what the price of natural gas could be at any given time is paramount, it could make or break economies. And in this case, as this project points out data-driven machine learning models deserve all the attention it could ever garner and even more.

**Future Enhancement**

# The project has been built using 2 models of prediction namely the Decision Tree method and Random Forest method with the accuracy score of over 97% on both the models (97.4% on Decision Tree and 97.74% on Random Forest Method). By doing some further research and learning the accuracy can be uplifted up to 100% which would be an ideal prediction real- time application which would be much more helpful in the trading sector.

**CHAPTER- 10**

**REFERENCES**

1. Ying, C.; Luh, P. B.; Che, G.; Yige, Z.; Michel, L. D.; Coolbeth, M. A.; Friedland, P. B.; Rourke, S. J. Short-Term Load Forecasting: Similar Day-Based Wavelet Neural Networks. IEEE Transactions on Power Systems, 2010, vol. 25, no. 1, pp. 322-330.

2. Feinberg, E. A.; Genethliou, D. Load Forecasting. in Applied Mathematics for Restructured Electric Power Systems, Chow, J.H., Wu, F.F., Momoh, J. Eds., Springer, US, 2005. pp 269-285

3. Moser, M., Jordan, M., Petsche, T. Eds., MIT Press, Cambridge, MA, 1996, pp. 281-287.

4. Felice, M. D.; Xin, Y. Short-Term Load Forecasting with Neural Network Ensembles: A Comparative Study [Application Notes]. Computational Intelligence Magazine: Springer Science, 2011, vol. 6, no. 3, pp. 47-56.

5. Park, D. C.; El-Sharkawi, M. A.; Marks, R. J.; Atlas, L. E.; Damborg, M. J. Electric load forecasting using an artificial neural network. IEEE Transactions on Power Systems, 1991, vol. 6, no. 2, pp. 442-449.

6. Liu, K.; Subbarayan, S.; Shoults, R. R.; Manry, M. T.; Kwan, C.; Lewis, F. I.; Naccarino, J. Comparison of very short-term load forecasting techniques. IEEE Transactions on Power Systems, 1996, vol. 19, no. 1, pp. 164-173.

7. Abdel-Aal, R. E. Short-term hourly load forecasting using abductive networks. IEEE Transactions on Power Systems, 2004, vol. 19, no. 1, pp. 164-173.

8. Fan, S.; Chen, C. Short-term load forecasting based on an adaptive hybrid method. IEEE Transactions on Power Systems, 2006, vol. 21, no. 1, pp. 392-401.

9. Buchananan, W.; Hodges, P.; Theis, J. Which way the natural gas price: an attempt to predict the direction of the natural gas spot price movements using trader positions. Energy Economics, 2001, vol. 23, pp. 279-293.

10. Nguyen, H. T.; Nabney, I. T.; Short-term electricity demand and gas price forecasts using wavelet transforms and adaptive models. Energy, 2010, vol. 35, no. 9, pp. 3674-3685.

11. Salehnia N. et al. Forecasting natural gas spot prices with nonlinear modeling using Gamma test analysis. J. of Nat.Gas Sci. and Eng., 2013, vol. 14, pp. 238-249.

12. Serletis, A.; Shah Moradi, A. Returns and volatility in the NYMEX Henry Hub natural gas futures market. OPEC Review: Energy Economics and related issues, 2006, pp. 171-186.