

**Emotion Detection Using CNN**

**A PROJECT REPORT**

***Submitted to***

**SRI VENKATESWARA UNIVERSITY COLLEGE OF ENGINEERING**

**In partial fulfilment of requirements for the award of the degree of**

**BACHELOR OF TECHNOLOGY IN**

**COMPUTER SCIENCE AND ENGINEERING**

**By**

**K ADITHYA (11716001)**

**P DHARANI DEVI (11716007) N HARSHITHA (11716014)**

***Under the guidance of***

**Prof.K.V Madhumurthy.,**

**Professor**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING SRI VENKATESWARA UNIVERSITY**

**COLLEGE OF ENGINEERING, TIRUPATI - 517502, 2020-21.**



**SRI VENKATESWARA UNIVERSITY COLLEGE OF ENGINEERING, TIRUPATI**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**CERTIFICATE**

**This is to certify that the project entitled “EMOTION DETECTION USING CNN” was genuine and has been carried out under my supervision in the Department of Computer Science and Engineering, Sri Venkateswara University College of Engineering. The work is comprehensive, complete and fit for evaluation carried out in partial fulfilment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering during the academic year 2020-21.**

**To the best of our knowledge matter embodied in the project has not been submitted to any other University/Institution for the award of any Degree or Diploma.**

**Guide: Head of the Department:**

Prof.K.V MADHUMURTHY Dr. P. VENKATA SUBBA REDDY

Professor, Department of CSE, Professor, Department of CSE, SVUCE, Tirupati. SVUCE, Tirupati.



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**SRI VENKATESWARA UNIVERSITY COLLEGE OF ENGINEERING, TIRUPATI – 517502.**

Declaration

**The project entitled “EMOTION DETECTION USING CNN” is a bona fide work performed by us, for the partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering from Sri Venkateswara University, Tirupati.**

**To the best of our knowledge matter embodied in the project has not been submitted to any other University/Institution for the award of any Degree or Diploma.**

**(K. ADITHYA 11716001)**

**(T. DHARANI DEVI 11716007)**

**(N. HARSHITHA 11716014)**



**Acknowledgement**

**We express our deep sense of gratitude and gratefulness to our respected guide, Prof.M.V MADHUMURTHY, Professor in Department of Computer Science and Engineering for the valuable help and guidance. We feel very much thankful to him for the encouragement he has given us in completing the project.**

**We are also grateful to our respected Dr. P. VENKATA SUBBA REDDY, Head of the department, CSE, for providing us valuable advices and reviews on the project and for permitting us to utilize all the necessary facilities of the institution.**

**We are also thankful to all the other faculty & staff members of our department for their kind co-operation and help.**

**Lastly, we would like to express our deep apperception towards our classmates and our indebtedness to our parents for providing us the moral support and encouragement.**

**(K. ADITHYA 11716001)**

**(T. DHARANI DEVI 11716007)**

**(N. HARSHITHA 11716014)**

# TABLE OF CONTENTS

### SYNOPSIS 8 – 10

1. **PREAMBLE 11 – 18**
   1. GENERAL INTRODUCTION
   2. STATEMENT OF PROBLEM
      1. EXISTING SYSTEM
      2. PROPOSED SYSTEM
   3. MOTIVATION OF THE PROJECT
   4. LITERATURE REVIEW
   5. HARDWARE & SOFTWARE REQUIREMENTS
   6. SOFTWARE REQUIREMENTS SPECIFICATION

### SYSTEM ANALYSIS 19 - 20

* 1. INTRODUCTION
  2. PROBLEM ANALYSIS
  3. FEASIBILITY STUDY

### SYSTEM DESIGN 21 - 32

* 1. INTRODUCTION
  2. SCOPE AND PURPOSE OF THIS DOCUMENT
  3. PROJECT TASKS
  4. DETAILED SYSTEM DESIGN
     1. MODULES
     2. SYSTEM DIAGRAM
     3. SYSTEM FLOWCHART
     4. DATASETS USED
     5. ALGORITHMS USED
     6. DATABASE TABLE
     7. SNAPSHOTS OF THE PROJECT

### TESTING 33 - 35

* 1. INTRODUCTION
  2. TYPES OF TESTING
  3. UNIT TESTING
  4. INTEGRATION TESTING
  5. ACCEPTANCE TESTING

### CODE SNIPPET 36 - 41

### SNAPSHOTS OF THE PROJECT 42 - 43

### CONCLUSION 44

### FUTURE ENHANCEMENT 45

### BIBLIOGRAPHY 46

# LIST OF FIGURES

1. SYSTEM DIAGRAM **23**
2. FLOWCHART OF TRAINING **24**
3. FLOWCHART OF TESTING **25**
4. IMAGE FROM FER2013 DATASET **27**
5. SNAPSHOT OF FER2013 DATASET **27**
6. HAAR-LIKE FEATURES **28**
7. APPLYING LBP OPERATION **30**
8. CIRCULAR LBP **31**

# CHAPTER 1 SYNOPSIS

## INTRODUCTION:

A Facial expression is the visible manifestation of the affective state, cognitive activity, intention, personality and psychopathology of a person and plays a communicative role in interpersonal relations. It has been studied for a long period of time and is obtaining the progress in recent decades.

## PROJECT OBJECTIVE:

The human face is main element to understand the individuals as well as gives the vital data, cutting-edge notion of user’s behavior via their one of kind expressions. Therefore, in biometric vicinity of the studies the Automatic Facial Expression Recognition has been one of the latest research topics.

* + The goal of facial expression recognition (FER) algorithms is to extract the discriminative and distinguishing features of a face.
  + Multiple methods have been devised which helps in identifying face and facial expression.
  + The various facial expressions are popularly identified on the basis of their geometric features, appearance features and hybrid features.

## PROBLEM DEFINITION:

As Robotics and Artificial intelligence is developing rapidly, interaction with the machines is becoming more and more important. In order to achieve that, detecting face and recognizing the facial expression becomes a vital and challenging task.

## EXISTING SYSTEM:

Facial expression recognition has been an active research area over the past few decades, and it is still challenging due to the high intra-class variation. Traditional approaches for this problem rely on hand-crafted features such as SIFT, HOG and LBP, followed by a

classifier trained on a database of images or videos. Most of these works perform reasonably well on datasets of images captured in a controlled condition, but fail to perform as good on more challenging datasets with more image variation and partial faces. In recent years, several works proposed an end-to-end framework for facial expression recognition, using deep learning models.

## PROPOSED SYSTEM:

Face and Emotion recognition can be performed using different features, such as face speech, and even text. Among these features, facial expressions are one of the most popular, if not the most popular, due to a number of reasons; they are visible, they contain many useful features for emotion recognition, and it is easier to collect a large dataset of faces ( other means for human recognition) Recently, with the use of deep learning and especially convolution neural networks (CNNs) , many features can be extracted and learned for a decent facial expression recognition system . It is, however, noteworthy that in the case of facial expressions, much of the clues come from a few parts of the face, e.g. the mouth and eyes, whereas other parts, such as ears and hair, play little part in the output.

This means that ideally, the machine learning framework should focus only on the important parts of the face, and less sensitive to other face regions. In this work we propose a deep learning-based framework for facial expression recognition, which takes the above observation into account, and uses attention mechanism to focus on the salient part of the face. We show that by using attentional convolutional network, even a network with few layers (less than 10 layers) is able to achieve very high accuracy rate. More specifically, this paper presents the following contributions:

* We propose an approach based on an attentional convolutional network, which can focus on feature-rich parts of the face, and yet, outperform remarkable recent works in accuracy.
* In addition, we use the visualization technique proposed in to highlight the face image’s most salient regions.

## SCOPE OF THE PROJECT:

Facial expression is one of the most common non-verbal ways that humans use to convey their internal emotional states and, consequentially, plays a significant role in

interpersonal interactions. Although there exists a wide range of possible facial expressions, psychologists have identified the seven basic ones (happiness, sadness, fear, disgust, surprise, anger and neutral) that are universally recognized (Izard 1971). It is straightforward that a system capable to perform an automatic recognition of the human emotions is a desirable task for a set of emerging applications.

# CHAPTER-2 PREAMBLE

## GENERAL INTRODUCTION:

Facial expressions play an important role in recognition of emotions and are used in the process of non-verbal communication. It gives us clue about the state of a person and enables to make conversation with the other person based on their mood. They are very important in daily emotional communication. They are also an indicator of feelings, allowing a man to express an emotional state. People, can immediately recognize an emotional state of a person. As a consequence, information on the facial expressions is often used in automatic systems of emotion recognition.

## STATEMENT OF PROBLEM:

### EXISTING SYSTEM:

Facial expression recognition has been an active research area over the past few decades, and it is still challenging due to the high intra-class variation. Traditional approaches for this problem rely on hand-crafted features such as SIFT, HOG and LBP, followed by a classifier trained on a database of images or videos. Most of these works perform reasonably well on datasets of images captured in a controlled condition, but fail to perform as good on more challenging datasets with more image variation and partial faces. In recent years, several works proposed an end-to-end framework for facial expression recognition, using deep learning models.

### PROPOSED SYSTEM:

Face and Emotion recognition can be performed using different features, such as face speech, and even text. Among these features, facial expressions are one of the most popular, if not the most popular, due to a number of reasons; they are visible, they contain many useful features for emotion recognition, and it is easier to collect a large dataset of faces ( other means for human recognition) Recently, with the use of deep learning and especially convolution neural networks (CNNs) , many features can be extracted and learned for a decent facial expression recognition system . It is, however, noteworthy that in the case of

facial expressions, much of the clues come from a few parts of the face, e.g. the mouth and eyes, whereas other parts, such as ears and hair, play little part in the output.

This means that ideally, the machine learning framework should focus only on the important parts of the face, and less sensitive to other face regions. In this work we propose a deep learning-based framework for facial expression recognition, which takes the above observation into account, and uses attention mechanism to focus on the salient part of the face. We show that by using attentional convolutional network, even a network with few layers (less than 10 layers) is able to achieve very high accuracy rate. More specifically, this paper presents the following contributions:

* We propose an approach based on an attentional convolutional network, which can focus on feature-rich parts of the face, and yet, outperform remarkable recent works in accuracy.
* In addition, we use the visualization technique proposed in to highlight the face image’s most salient regions.

## MOTIVATION OF THE PROJECT:

In the beginning, facial expression analysis was essentially a research topic for psychologists. However, recent progresses in image processing and pattern recognition have motivated significant research works on automatic facial expression recognition. The facial expressions are useful for efficient interaction. Face recognition is important for the interpretation of facial expressions in applications such as intelligent, man-machine interface and communication, intelligent visual surveillance, teleconference and real-time animation from live motion images. As a result, we are highly motivated to develop a system that recognizes facial expression.

## LITERATURE REVIEW:

Rajesh Kumar G A, Ravi Kant Kumar, Goutam Sanyal [1] contributed on “Facial Emotion Analysis using Deep Convolution Neural Network”, 2017. They proposed that, human emotions are mental states of feelings that arise spontaneously rather than through conscious effort and are accompanied by physiological changes in facial muscles which implies expressions on face. Some of critical emotions are happy, sad, anger, disgust, fear, surprise etc. Facial expressions play a key role in nonverbal communication which appears due to internal feelings of a person that reflects on the faces. In order to computer modelling of human’s emotion, a plenty of research has been accomplished. But still it is far behind

from human vision system. In this system, they are providing better approach to predict human emotions (Frames by Frames) using deep Convolution Neural Network (CNN) and how emotion intensity changes on a face from low level to high level of emotion. In this algorithm, FERC-2013 database has been applied for training. The assessment through the proposed experiment confers quite good result and obtained accuracy may give encouragement to the researchers for future model of computer-based emotion recognition system.

Viola, Paul, and Michael Jones [2], researched on, “Rapid object detection using a boosted cascade of simple features”, 2001. They worked on the machine learning approach for visual object detection which is capable of processing images extremely rapidly and achieving high detection rates. It is distinguished by three key contributions. The first is the introduction of a new image representation called the “Integral image” which allows the features used by our detector to be computed very quickly. The second is a learning algorithm, based on AdaBoost, which selects a small number of critical visual features from a larger set and yields extremely efficient classifiers. The third contribution is a method for combining increasingly more complex classifiers in a “cascade” which allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions. The cascade can be viewed as an object specific focus-of-attention mechanism which unlike previous approaches provides statistical guarantees that discarded regions are unlikely to contain the object of interest, In the domain of face detection the system yields detection rates comparable to the best previous systems. Used in real-time applications, the detector runs at 15 frames per second without resorting to image differencing or skin color detection.

Lienhart, Rainer, and Jochen Maydt [3] gave the work, "An extended set of Haar-like features for rapid object detection”, 2002. Recently Viola et al. have introduced a rapid object detection scheme based on a boosted cascade of simple feature classifiers. In this study they introduced a novel set of rotated Haar-like features. These novel features significantly enrich the simple features and can also be calculated efficiently. With these new rotated features, the sample face detector shows off on average a 10% lower false alarm rate at a given hit rate. They also present a novel post optimization procedure for a given boosted cascade improving on average the false alarm rate further by 12.5%.

Wei-feng Liu, Shu-juan Li, Yan-jiang Wang [4] proposed “Automatic facial expression recognition based on Local Binary Patterns of Local Areas”, 2009. They stated

that, Automatic facial expression recognition is one of the great challenges in facial expression recognition field. An algorithm of automatic facial expression recognition is proposed based on Local Binary Patterns of local areas (LLBP)in this work. First, the position of eye balls is fixed by projection method. Then the local areas of the eyes and mouth’s neighborhood could be determined through the prior knowledge of face structure. The LBP feature on the local areas is then computed as the facial feature for facial expression recognition. Finally, the recognition experiment is conducted on the JAFFE facial database, which showed the reliability of the method proposed.

Saad Albawi, Tareq Abed Mohammed, Saad Al-Zawi [5] contributed on, “Understanding of a Convolutional Neural Network”, 2017. According to them, the term Deep Learning or Deep Neural Network refers to Artificial Neural Networks (ANN) with multi layers. Over the last few decades, it has been considered to be one of the most powerful tools, and has become very popular in the literature as it is able to handle a huge amount of data. The interest in having deeper hidden layers has recently begun to surpass classical methods performance in different fields; especially in pattern recognition. One of the most popular deep neural networks is the Convolutional Neural Network (CNN). It takes this name from mathematical linear operation between matrixes called convolution. CNN have multiple layers; including convolutional layer, non-linearity layer, pooling layer and fully connected layer. The convolutional and fully- connected layers have parameters but pooling and non- linearity layers don't have parameters. The CNN has an excellent performance in machine learning problems. Specially the applications that deal with image data, such as largest image classification data set (Image Net), computer vision, and in natural language processing (NLP) and the results achieved were very amazing. In this study, they will explain and define all the elements and important issues related to CNN, and how these elements work. In addition, they also state the parameters that effect CNN efficiency. They assumed that the readers have adequate knowledge about both machine learning and artificial neural network.

Zhang, Zhengyou, et al., [6] worked on "Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron”, 1998. They investigate the use of two types of features extracted from face images for recognizing facial expressions. The ﬁrst type is the geometric positions of a set of ﬁducial points on a face. The second type is a set of multi-scale and multi-orientation Gabor wavelet coefficients extracted from the face image at the ﬁducial points. They can be used either independently or jointly. The architecture they developed is based on a two-layer perceptron. The recognition

performance with different types of features has been compared, which shows that Gabor wavelet coefﬁcients are much more powerful than geometric positions. Furthermore, since the ﬁrst layer of the perceptron actually performs a nonlinear reduction of the dimensionality of the feature space, they have also studied the desired number of hidden units, i.e., the appropriate dimension to represent a facial expression in order to achieve a good recognition rate. It turns out that ﬁve to seven hidden units are probably enough to represent the space of feature expressions.

XueMei Zhao, ChengBing Wei [7] proposed “A Real-time Face Recognition System Based on the Improved LBPH Algorithm”. They found that The Local Binary Pattern Histogram (LBPH)algorithm is a simple solution on face recognition problem, which can recognize both front face and side face. However, the recognition rate of LBPH algorithm under the conditions of illumination diversification, expression variation and attitude deflection are decreased. To solve this problem, a modified LBPH algorithm based on pixel neighborhood greymedian (MLBPH) is proposed. The grey value of the pixel is replaced by the median value of its neighborhood sampling value, and then the feature value is extracted by the sub blocks and the statistical histogram is established to form the MLBPHfeaturedictionary, which is used to recognize the human face identity compared with test image. Experiments are carried on FERET standard face database and the creation of new face database, and the results show that MLBPH algorithm is superior to LBPH algorithm in recognition rate.

Mostafa Mohammadpour, Seyyed Mohammad.RHashemi [8] gave “Facial Emotion Recognition using Deep Convolutional Networks” stated that Facial emotion recognition is an emerging field which use in many now a day’s application including social robots, neuro marketing and games. Non-verbal communication methods like facial expressions, eye movement and gestures are used in many applications of human computer interaction, which among them facial emotion is widely used because it conveys the emotional states and feelings of persons. The emotion recognition is not an easy task because there is no landmark distinction between the emotions on the face and also there are a lot of complexity and variability. In the traditional machine learning algorithm, some important extracted features used for modelling the face, so, it cannot achieve high accuracy rate for recognition of emotion because the features are hand-engineered and depend on prior knowledge. Convolutional neural networks (CNN) have developed in this work for recognition facial emotion expression and classify the min to seven basic categories. Instead of calculating hand

engineered features, CNN calculates features by learning automatically. Then owelty of the proposed method is using facial action units (AUs) of the face which first these units are recognized by CNN and in corporate to recognizing these basic emotion states. To evaluated the proposed model, Cohn-Kanade database is used so that the model achieves the best accuracy rate 97.01 by incorporating AU while other works in the literature used a direct CNN and achieve accuracy rate 95.75.

To summarize from the above study, each of the techniques or the methods used consists its own flaws in it. Comparatively, the deep CNN which has been proposed found to be the best. It is considered to be the best method to recognize the facial expressions.

## HARDWARE & SOFTWARE REQUIREMENTS:

#### Software Requirement:

Following are the software requirement necessary of the project:

* + - 1. IDLE (Python 3.6 64-bit)
      2. Java Development Kit
      3. OpenCV, Keras, TensorFlow and Tkinter libraries.

#### Hardware Requirement:

Following are the hardware requirement that is most important for the project:

* + - 1. Fluently working Laptops
      2. RAM minimum 4Gb
      3. Web Camera.

## SOFTWARE REQUIREMENTS SPECIFICATION:

#### Introduction to Python:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity

and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. Python is considered an interpreted language because Python programs are executed by an interpreter.

#### Keras:

Keras is an [open-source](https://en.m.wikipedia.org/wiki/Open-source_software) [neural-network](https://en.m.wikipedia.org/wiki/Artificial_neural_network) library written in [Python.](https://en.m.wikipedia.org/wiki/Python_(programming_language)) It is capable of running on top of [TensorFlow,](https://en.m.wikipedia.org/wiki/TensorFlow) [Microsoft Cognitive Toolkit,](https://en.m.wikipedia.org/wiki/Microsoft_Cognitive_Toolkit) [Theano](https://en.m.wikipedia.org/wiki/Theano_(software)), or [PlaidML.](https://en.m.wikipedia.org/wiki/PlaidML) Designed to enable fast experimentation with [deep neural networks](https://en.m.wikipedia.org/wiki/Deep_learning), it focuses on being user-friendly, modular, and extensible. Keras was conceived to be an interface rather than a standalone machine-learning framework. It offers a higher-level, more intuitive set of abstractions that make it easy to develop deep learning models. Keras contains numerous implementations of commonly used neural-network building blocks such as layers, [objectives,](https://en.m.wikipedia.org/wiki/Objective_function) [activation](https://en.m.wikipedia.org/wiki/Activation_function) [functions,](https://en.m.wikipedia.org/wiki/Activation_function) [optimizers,](https://en.m.wikipedia.org/wiki/Mathematical_optimization) and a host of tools to make working with image and text data easier. In addition to standard neural networks, Keras has support for [convolutional](https://en.m.wikipedia.org/wiki/Convolutional_neural_networks) and [recurrent](https://en.m.wikipedia.org/wiki/Recurrent_neural_networks) [neural networks](https://en.m.wikipedia.org/wiki/Recurrent_neural_networks). Keras allows users to productize deep models on smartphones ([iOS](https://en.m.wikipedia.org/wiki/IOS) and [Android](https://en.m.wikipedia.org/wiki/Android_(operating_system))), on the web, or on the [Java Virtual Machine.](https://en.m.wikipedia.org/wiki/Java_Virtual_Machine)

#### TensorFlow:

Tensor Flow is a free and open source software library for dataflow and differentiable programming across a range of tasks. It is symbolic math library, and is also used for machine learning applications such as neural network.

TensorFlow object recognition algorithms classify and identify arbitrary objects within larger images. This is usually used in engineering applications to identify shapes for modelling purposes (3D space construction from 2D images) and by social networks for photo tagging (Facebook’s Deep Face).

#### Tkinter:

Tkinter is [free software](https://en.wikipedia.org/wiki/Free_software) released under a [Python license.](https://en.wikipedia.org/wiki/Python_license) Tkinter is Python's de-facto standard GUI (Graphical User Interface) package. Tkinter is not the only [GUI](https://wiki.python.org/moin/GuiProgramming) [Programming](https://wiki.python.org/moin/GuiProgramming) toolkit for Python.

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object- oriented interface to the Tk GUI toolkit. Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets. The Button widget is used to display buttons in your application. The Entry widget is used to display a single-line text field for accepting values from a user.

### IDLE:

IDLE is integrated development and learning environment for editing and running Python 2.x or Python 3 programs. IDLE is an integrated development environment for Python, which has been bundled with the default implementation of the language. The IDLE GUI is automatically installed with the Python interpreter. IDLE was designed specifically for use with Python. IDLE has a number of features which helps to develop Python programs including powerful syntax highlighting.

* 1. **INTRODUCTION:**

# CHAPTER-3 SYSTEM ANALYSIS

System Analysis is a process of collecting factual data, understand the processes involved, identifying problems and recommending feasible suggestions for improving the system functioning. This involves studying the business processes, gathering operational data, understand the information flow, finding out bottlenecks and evolving solutions for overcoming the weakness of the system so as to achieve the organizational goals. System Analysis also includes subdividing of complex process involving the entire system, identification of data store and manual processes.

The major objectives of system analysis are to find answers for each business process, what is being done, how is it being done, who is doing it, when is he doing it, why is it being done and how is it being done, who is doing it, when is he doing it, why is it being done and how can it be improved? It attempts to give birth to a new efficient system that satisfies the current needs of the user and has scope for future growth within the organizational constraints. The result of this process is logical system design. System analysis is an iterative process that continues until a preferred and acceptable solution emerges.

## PROBLEM ANALYSIS:

#### Determines the present position:

The main objective of our project is to fulfil the student requirement by providing resources according to their curriculum.

#### Understanding student’s requirements:

Provides required resources to the students, which helps them to increase their understanding level.

## FEASIBILITY STUDY:

In case the system proposal is acceptable to the management, the next phase is to examine the feasibility of the system. The feasibility study is basically the test of the proposed system in the light of its workability, meeting user’s requirements, effective use of resources. These are categorized as technical, operational, economical and schedule

feasibility. The main goal of feasibility study is not to solve the problem but to achieve the scope. The result is a feasibility report submitted to the management. This may be accepted or rejected. The system cycle proceeds only if the management accepts it.

# CHAPTER-4 SYSTEM DESIGN

## INTRODUCTION:

Based on the user requirements and the detailed analysis of the existing system the new system must be designed. This is the phase of system designing. It is the most crucial phase in the developments of a system. The logical system design arrived as a result of systems analysis is converted into physical system design. Normally, the design proceeds in two stages:

* + 1. Preliminary or General Design,
    2. Structured or Detailed Design.

## SCOPE AND PURPOSE OF THIS DOCUMENT:

* The system can be used in mini-marts, shopping centre to view the feedback of the customers to enhance the business.
* This system can be used for lie detection amongst criminal suspects during interrogation.
* The system can also be used for educational purpose, one can get feedback on how the student is reacting during the class.

## PROJECT TASK:

#### What needs are being fulfilled?

We fulfill the needs of the client by recognizing the face and identifying the exact expression based on their geometric features of the face.

#### How is it useful for them?

It plays an important role in recognizing the emotions and are used as non-verbal communication.

It gives them clue about the state of a person and enables to make conversation with the other person based on their mood.

This is also an indicator of feelings, allowing a man to express an emotional state.

## DETAILED DESIGN:

### MODULES:

Modules used in our project are,

* + - 1. Face capturing module
      2. Preprocessing module
      3. Training module
      4. Face recognition module
      5. Expression recognition module

#### Face Capturing Module:

* + In this phase we are capturing the faces of people for future processing using web camera or external web camera.
  + Without capturing the image none of the process can be done and there is no chance of detecting the emotions.

#### Preprocessing Module:

* + After capturing images, we will preprocess the images.
  + In this phase we will convert the color images to gray scale images.

#### Training Module:

* + In this phase we will prepare a dataset that is binary array of all captured images.
  + The captured images will store in .YML file, which stores all facial data.
  + .YML file helps us to speedup processing of the captured images.

#### Face Recognition Module:

* + The initial step of face recognition process is to train the facial data to the host system.
  + Using the web camera of the computer system, 60 distinct images of the face is taken.
  + In this module we will recognize the faces using LBPH algorithm. LBPH is an acronym for local binary pattern histogram. It will recognize the faces with the face ID and NAME that are all previously stored.

#### Face Expression Recognition Module:

* + Facial expression recognition software is a technology which uses biometric markers to detect emotions in human faces.
  + It extracts and analyzes information from an image it is able to deliver unfiltered, unbiased, emotional response or data.

### SYSTEM DIAGRAM:



**Webcam**

**Train**

**[LBPH]**

**Trained**

**Dataset**

Load

Trainer.yml face.xml

**FER2013**

**Dataset**

Load

**Class1:**

**Angry**

**Class2:**

**Disgust**

**Class3:**

**Scared**

**Class4:**

**Happy**

**Class5:**

**Sad**

**Class6:**

**Surprised**

**Class7:**

**Neutral**

**7 Way Output**

**Deep Convolutional Neural network**

**Face Detection**

**Image Pre- processing**

**Capture Face**

#### Fig 3.1 System Diagram

A webcam is used to capture, detect and recognize the expressions of a person. Rectangular frame on the face region is obtained in the camera; this detection of face region from a non-face region is done by making use of Viola Jones algorithm, LBPH Face Recognizer algorithm with a Haarcascade frontal face dataset. Around 60 Images of a person

are captured, preprocessed and stored in a folder with his/her ID and Name (Labels are stored in face.csv file). These images are trained using LBPH algorithm and the trained dataset is stored as Trainer.yml. During Face Detection: Pretrained dataset is used to match the face in video camera. If person’s face matches with that of trained dataset, then his/her ID and Name is shown. The classification of the acquired face is done using convolutional neural network by making use of FER2013 database. The highest probability of expression acquired with respect to the features, is shown as the facial expression. One out of seven expressions are displayed with the detected image of the person.

### SYSTEM FLOWCHART:

Viola Jones Algorithm



**Start**

**Detect the Face**

**Image pre- processing**

**Train [LBPH]**

Trainer.yml face.xml



**Trained Dataset**

**Fig. Flowchart of Training**



**Start**

**Cropped Face**

**Trained**

**Dataset**

load

**FER2013**

**Dataset**

load

**Deep Convolutional Neural Network**

**Face Detection**

**Recognition Result**

**Fig. Flowchart of Testing/Prediction**

* + 1. **Datasets Used:**

Some of the public databases used to evaluate the facial expression recognition algorithms:

#### Haarcascade frontal face dataset:

For the purpose of detection of face from images using features, Haar Cascade Classifier is used. The haarcascade\_frontalface\_default.xml [17] is used to detect the frontal face. It was developed by Viola and Jones on the basis of the proposal by Papa Georgiou et. al in 1998[2].

To ensure that the extracted faces are positioned in the same location, we have used an additional classifier, from the same OpenCV library, called 'haarcascade\_eye.xml'[18]. This detects the region of the eyes, which is then used to adjust the left and right margins of the face window, to ensure equal distance between eyes and the sides of the face. In this way, unnecessary information (such as hair, ears, background) is discarded and the extracted faces will have normalized positions.

#### FER2013 dataset:

FER2013[15] is an open-source dataset which is first created for an ongoing project by Pierre-Luc Carrier and Aaron Courville, then shared publicly for a Kaggle competition [15]. The FER2013 database was introduced during the ICML 2013 Challenges in Representation Learning. FER2013 is a large-scale and unconstrained database collected automatically by the Google image search API. All images have been registered and resized to 48\*48 pixels after rejecting wrongfully labelled frames and adjusting the cropped region. This dataset consists of 35,887 grayscale, 48x48 sized face images with various emotions -7 emotions, all labelled-.

Emotion labels in the dataset:

**0:** -4593 images- *Angry*

**1:** -547 images- *Disgust*

**2:** -5121 images- *Fear*

**3:** -8989 images- *Happy*

**4:** -6077 images- *Sad*

**5:** -4002 images- *Surprise*

**6:** -6198 images- *Neutral*

During the competition, 28,709 images and 3,589 images were shared with the participants as training and public test sets respectively and the remaining 3,589 images were kept as private test set to find the winner of the competition. The dataset was set to accessible to everyone after completing the competition.

The FER-2013 dataset was created by gathering the results of a Google image search of each emotion and synonyms of the emotions. The images in FER-2013[15] consist of both posed and un-posed headshots.



**Fig.** Example images from the FER2013 dataset.

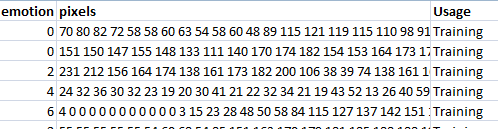
Figure illustrating variability in illumination, age, pose, expression intensity, and occlusions that occur under realistic conditions. Images in the same column depict identical expressions, namely anger, disgust, fear, happiness, sadness, surprise, as well as neutral.

The data file contains 3 columns — Class, Image data, and Usage.

1. Emotion class: is a digit between 0 to 6 and represents the emotion depicted in the corresponding picture. Each emotion is mapped to an integer as shown below.

0- 'Angry’ 1-'Disgust’ 2-'Fear’ 3-'Happy’ 4-'Sad’ 5-'Surprise’ 6-'Neutral'

1. Image data: is a string of 2,304 numbers and these are the pixel intensity values of our image, we will cover this in detail in a while.
2. Usage: It denotes whether the corresponding data should be used to train the network or test it.



**Fig. A Snapshot of FER2013 dataset**

### ALGORITHM USED:

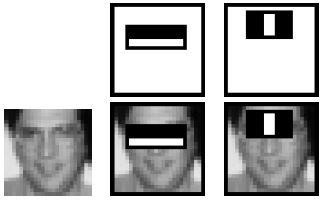
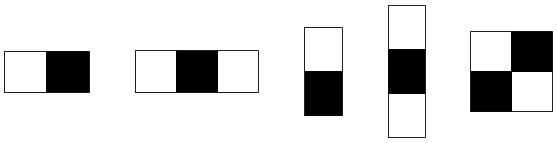
#### Viola Jones Algorithm:

A Haar-like feature considers neighboring rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. This difference is then used to categorize subsections of an image. In the detection of human faces, Commonly, the areas around the eyes are darker than the areas on

the cheeks. Haar-like feature for face detection is therefore a set of two neighboring rectangular areas above the eye and cheek regions. The algorithm [2] has four stages:

* + - 1. Haar Feature Selection
      2. Creating Integral Images
      3. Ad boost Training
      4. Cascading Classifiers

It is well known for being able to detect faces and body parts in an image, but can be trained to identify almost any object.



1. The 5 types of Haar-like templates; the value of each rectangle feature is computed by subtracting the sum of the black area, from the white area.
2. Method of applying the rectangle features on the 24x24 pixels image of the face.

#### Fig. Haar-like features

First step is to collect the Haar Features [3]. A Haar feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. An integral image is summed-area table is a data structure and algorithm for quickly and efficiently generating the sum of values in a rectangular subset of a grid. Hence integral images are used to make the detection super-fast. To select the best features out of 160000+ features. A concept called Ad boost is used which both selects the best features and trains the classifiers that use them. Cascade Classifier is the collection of phases. At each phase, it is determined whether the given sub window contains a face or a non-face. If it fails then it is considered as a non-face.

The characteristics of Viola–Jones algorithm which make it a good detection algorithm are:

* 1. Robust – very high detection rate (true-positive rate) & very low false-positive rate always.
  2. Real time – For practical applications at least 2 frames per second must be processed.
  3. Face detection only (not recognition) - The goal is to distinguish faces from non-faces (detection is the first step in the recognition process).

#### Local Binary Pattern (LBP):

Local Binary Pattern (LBP) [4] is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. It was first described in 1994 (LBP) and has since been found to be a powerful feature for texture classification. It has further been determined that when LBP is combined with histograms of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets.

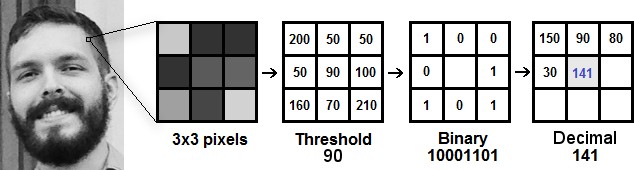
Using the LBP combined with histograms the face images are represented with a simple data vector. As LBP is a visual descriptor it can also be used for face recognition tasks, as can be seen in the following step-by-step explanation [14].

The steps of the algorithm:

1. **Parameters**: The LBPH uses 4 parameters:
   1. **Radius**: the radius is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1.
   2. **Neighbors**: the number of sample points to build the circular local binary pattern. Keep in mind: the more sample points you include, the higher the computational cost. It is usually set to 8.
   3. **Grid X**: the number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.
   4. **Grid Y**: the number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.
2. **Training the Algorithm**: First, train the algorithm. To do so, a dataset with the facial images of the people that are to be recognize is used. An ID is set (it may be a number or the name of the person) for each image, so the algorithm will use this information to

recognize an input image and give you an output. Images of the same person must have the same ID. With the training set already constructed, LBPH computational steps are

1. **Applying the LBP operation**: The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameter’s **radius** and **neighbors’** image below shows this procedure:

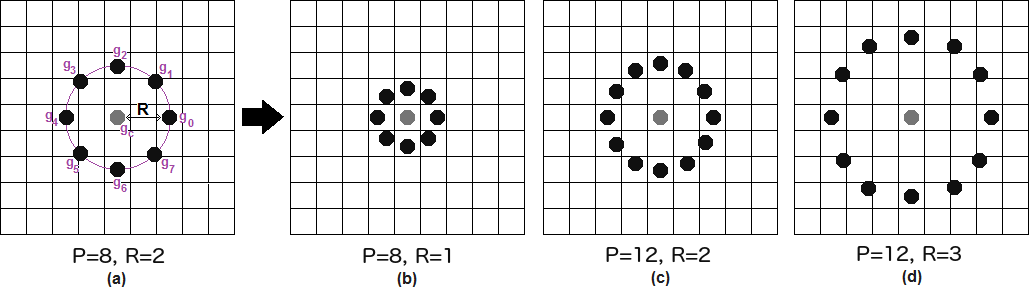


#### Fig 3.7Applying LBP operations

Steps followed here are:

* 1. Get a facial image in grayscale and a part of this image as a window of 3x3 pixels.
  2. It can also be represented as a 3x3 matrix containing the intensity of each pixel (0~255).
  3. Then, take the central value of the matrix to be used as the threshold. This value will be used to define the new values from the 8 neighbors.
  4. For each neighbor of the central value (threshold), a new binary value is set. Set 1 for values equal or higher than the threshold and 0 for values lower than the threshold. Now, the matrix will contain only binary values (ignoring the central value). It is required to concatenate each binary value from each position from the matrix line by line into a new binary value (e.g. 10001101).
  5. At the end of this procedure (LBP procedure), a new image appears, which represents better the characteristics of the original image.

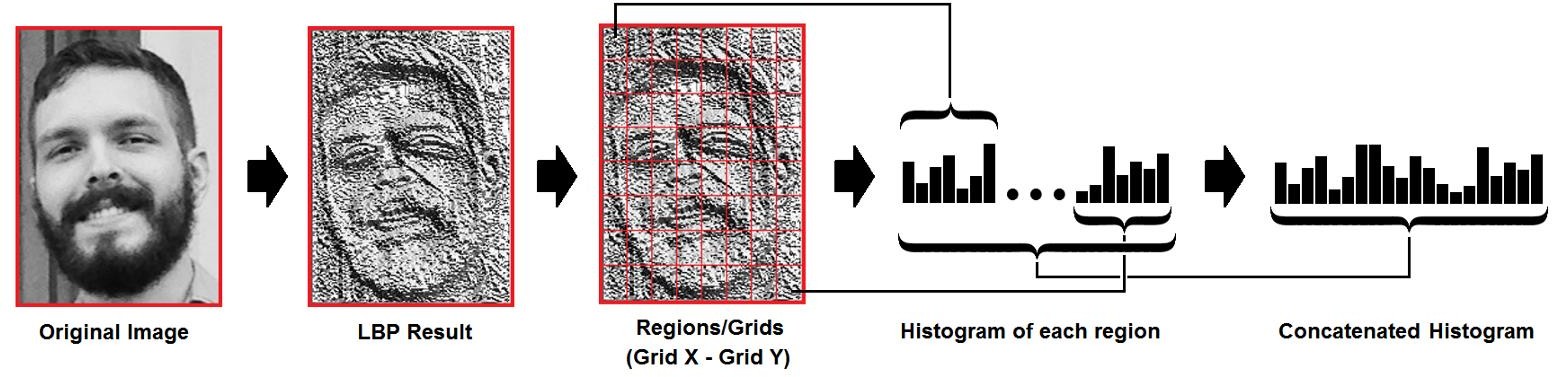
The LBP procedure was expanded to use a different number of radius and neighbors, it is called Circular LBP.



#### Fig 3.8 Circular LBP

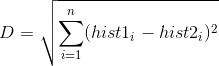
It can be done by using **bilinear interpolation**. If some data point is between the pixels, it uses the values from the 4 nearest pixels (2x2) to estimate the value of the new data point.

1. **Extracting the Histograms**: Now, using the image generated in the last step, the **Grid X** and **Grid Y** parameters to divide the image into multiple grids can be used, as can be seen in the following image:



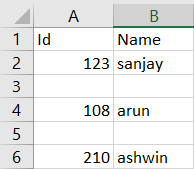
#### Fig 3.9 Haar-like features

Based on the image above, the histogram of each region is extracted as follows: As the image is in grayscale, each histogram (from each grid) will contain only 256 positions (0~255) representing the occurrences of each pixel intensity. Each histogram is concatenated to create a new and bigger histogram. Supposing if there is an 8x8 grid, in final histogram there will be 8x8x256=16.384 positions. The final histogram represents the characteristics of the image original image.

1. **Performing the face recognition**: In this step, the algorithm is already trained. Each histogram created is used to represent each image from the training dataset. So, given an input image, it performs the steps again for this new image and creates a histogram which represents the image.
   1. So, to find the image that matches the input image comparison of two histograms is needed and return the image with the closest histogram.
   2. Various approaches are used to compare the histograms (calculate the distance between two histograms), for example: **Euclidean distance**, **chi-square**, **absolute value**, etc. In this example, the Euclidean distance can be used (which is quite known) based on the following formula:
   3. A threshold and the ‘confidence’ to automatically estimate if the algorithm has correctly recognized the image is used. It is assumed that the algorithm has successfully recognized if the confidence is lower than the threshold defined.

### DATABASE TABLE:

A backup database is created which contains the essential information required for system to work. Database table contains two columns which stores ID and NAME are as given below,



* 1. **INTRODUCTION:**

# CHAPTER-5 TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

## TYPES OF TESTS:

### FUNCTIONAL TESTING:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

**Valid Input :** identified classes of valid input must be accepted. **Invalid Input:** identified classes of invalid input must be rejected. **Functions :** identified functions must be exercised.

**Output :** identified classes of application outputs must be exercised.

**Systems/Procedures :** interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

### SYSTEM TESTING:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

### WHITE BOX TESTING:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

### BLACK BOX TESTING:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

### UNIT TESTING:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

#### Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

#### Test objectives

* + - All field entries must work properly.
    - Pages must be activated from the identified link.
    - The entry screen, messages and responses must not be delayed.

#### Features to be tested

* + - Verify that the entries are of the correct format
    - No duplicate entries should be allowed
    - All links should take the user to the correct page.

### INTEGRATION TESTING:

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications,

e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

### ACCEPTANCE TESTING:

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

# CHAPTER-6 CODE SNIPPET

**def TakeImages ():** Id=(txt.get())#10001 name=(txt2.get())#Naveen if(Id.isdigit() and name.isalpha()):

cam = cv2.VideoCapture(0)#it will open the camera window in video mode a = "haarcascade\_frontalface\_default.xml" detector=cv2.CascadeClassifier(a)

sampleNum=0 while(True):

ret, img = cam.read()

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY) faces = detector.detectMultiScale(gray, 1.3, 5)

for (x,y,w,h) in faces: cv2.rectangle(img,(x,y),(x+w,y+h),(255,100,50),2) #incrementing sample number sampleNum=sampleNum+1

#saving the captured face in the dataset folder TrainingImage

cv2.imwrite("TrainingImage\ "+name +"."+Id +'.'+ str(sampleNum) + ".jpg", gray[y:y+h,x:x+w])

#display the frame cv2.imshow('frame',img)

#wait for 100 miliseconds

if cv2.waitKey(100) & 0xFF == ord('q'): break

# break if the sample number is morethan 100 elif sampleNum>60:

break cam.release()

cv2.destroyAllWindows()

res = "Images Saved for ID : " + Id +" Name : "+ name row = [Id , name]

with open('StudentDetails\StudentDetails.csv','a+') as csvFile: writer = csv.writer(csvFile)

writer.writerow(row) csvFile.close() message.configure(text= res)

else:

if(Id.isdigit()):

res = "Enter Alphabetical Name" message.configure(text= res)

if(name.isalpha()):

res = "Enter Numeric Id" message.configure(text= res)

#### def TrainImages():

recognizer=cv2.face\_LBPHFaceRecognizer.create()#recognizer= cv2.face.LBPHFaceRecognizer\_create()#$cv2.createLBPHFaceRecognizer()

a = "haarcascade\_frontalface\_default.xml" detector =cv2.CascadeClassifier(a)

faces,Id = getImagesAndLabels("TrainingImage") recognizer.train(faces, np.array(Id)) recognizer.save("TrainingImageLabel\Trainner.yml") res = "Image Trained"#+",".join(str(f) for f in Id) message.configure(text= res)

#### def TrackImages():

recognizer = cv2.face.LBPHFaceRecognizer\_create()#cv2.createLBPHFaceRecognizer()

recognizer.read("TrainingImageLabel\Trainner.yml") a = "haarcascade\_frontalface\_default.xml" faceCascade = cv2.CascadeClassifier(a); df=pd.read\_csv("StudentDetails\StudentDetails.csv") cam = cv2.VideoCapture(0)

font = cv2.FONT\_HERSHEY\_SIMPLEX #font family.. while True:

ret, im =cam.read() gray=cv2.cvtColor(im,cv2.COLOR\_BGR2GRAY) faces=faceCascade.detectMultiScale(gray, 1.2,5) for(x,y,w,h) in faces:

cv2.rectangle(im,(x,y),(x+w,y+h),(225,0,0),2) Id, conf = recognizer.predict(gray[y:y+h,x:x+w]) if(conf < 50):

aa=df.loc[df['Id'] == Id]['Name'].values tt=str(Id)+"-"+aa #1001-Kiran

else:

Id='Unknown' tt=str(Id)

cv2.putText(im,str(tt),(x,y+h), font, 1,(255,255,255),2) ret,frame = cam.read()

#reading the frame

frame = imutils.resize(frame,width=300)

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY) faces =

face\_detection.detectMultiScale(gray,scaleFactor=1.1,minNeighbors=5,minSize=(30,30),flag s=cv2.CASCADE\_SCALE\_IMAGE)

canvas = np.zeros((250, 300, 3), dtype="uint8") frameClone = frame.copy()

if len(faces) > 0:

faces = sorted(faces, reverse=True, key=lambda x: (x[2] - x[0]) \* (x[3] - x[1]))[0] (fX, fY, fW, fH) = faces

# Extract the ROI of the face from the grayscale image, resize it to a fixed 28x28 pixels, and then prepare

# the ROI for classification via the CNN roi = gray[fY:fY + fH, fX:fX + fW]

roi = cv2.resize(roi, (64, 64)) roi = roi.astype("float") / 255.0 roi = img\_to\_array(roi)

roi = np.expand\_dims(roi, axis=0)

preds = emotion\_classifier.predict(roi)[0] emotion\_probability = np.max(preds) label = EMOTIONS[preds.argmax()]

for (i, (emotion, prob)) in enumerate(zip(EMOTIONS, preds)): # construct the label text

text = "{}: {:.2f}%".format(emotion, prob \* 100) w = int(prob \* 300)

cv2.rectangle(canvas, (7, (i \* 35) + 5),

(w, (i \* 35) + 35), (255, 0,0), -1)

cv2.putText(canvas, text, (10, (i \* 35) + 23),

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

cv2.putText(frameClone, label, (fX, fY - 10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.45, (255, 0, 0), 2)

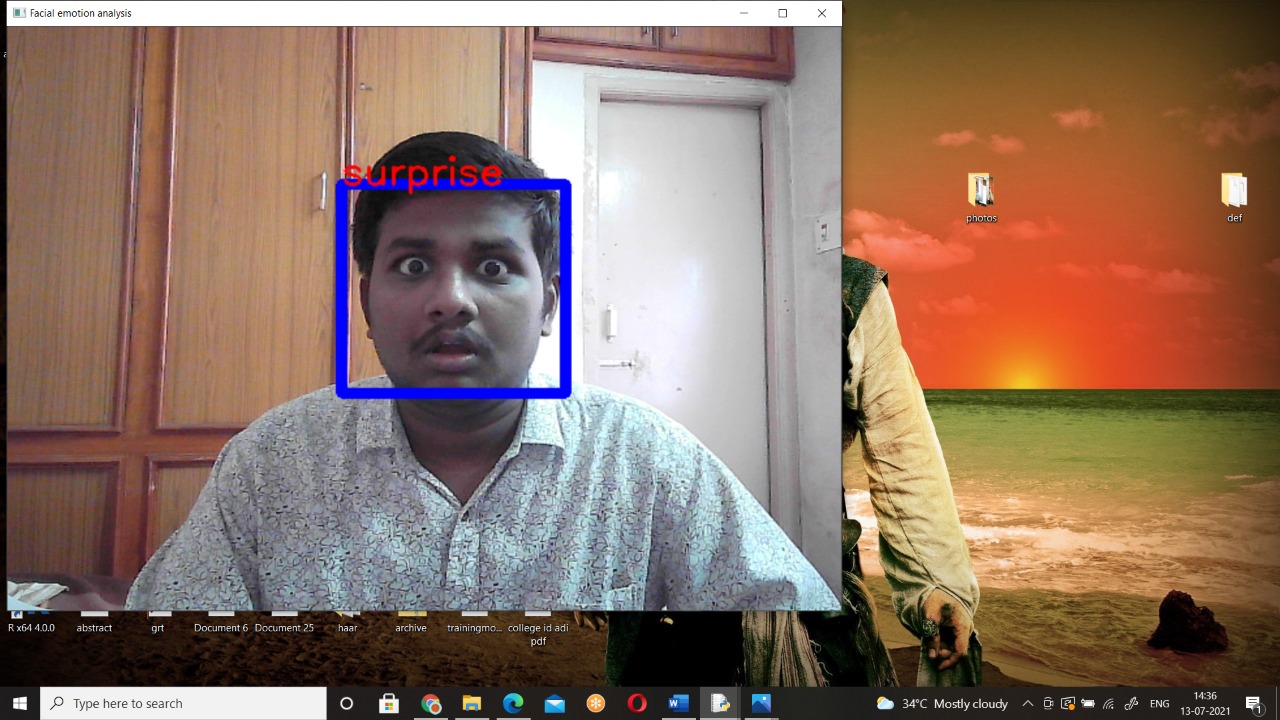
cv2.rectangle(frameClone, (fX, fY), (fX + fW, fY + fH),(255, 0, 0), 2) cv2.imshow('im',im)

cv2.imshow('emotions', frameClone) cv2.imshow("Prob Ranges", canvas)

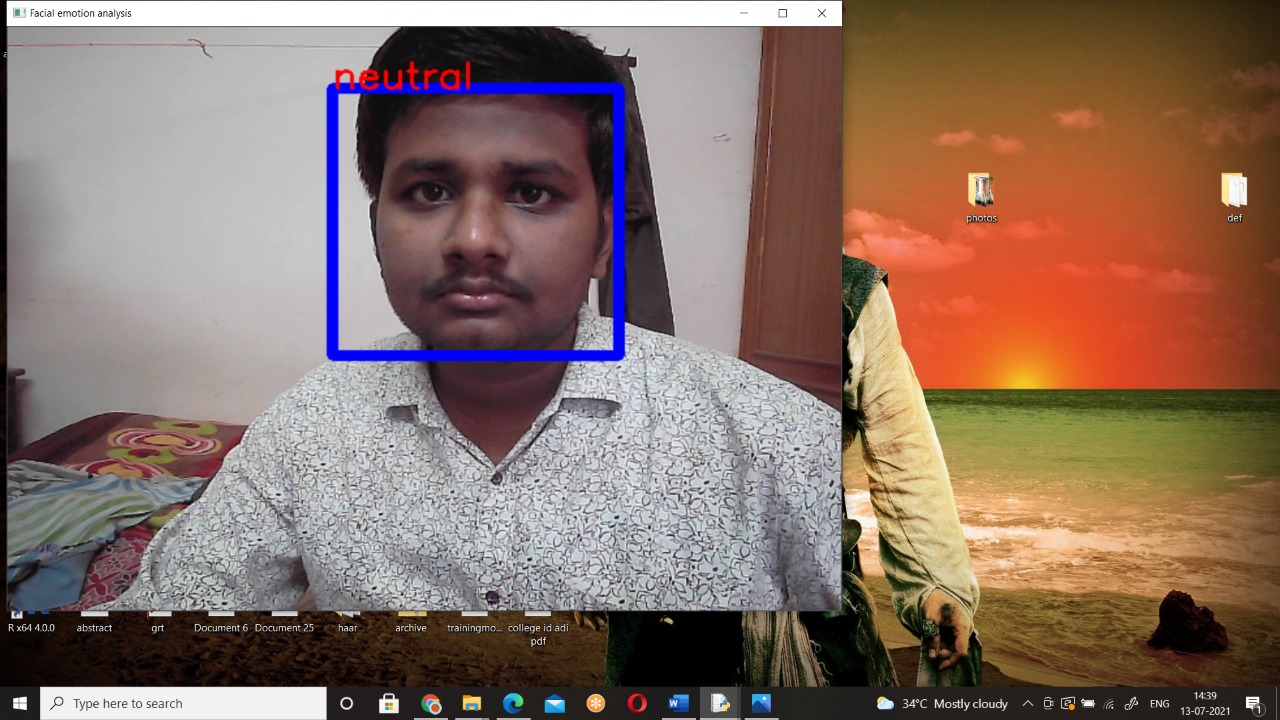
if (cv2.waitKey(1)==ord('q')): break

cam.release() cv2.destroyAllWindows()

# CHAPTER-7 SNAPSHOTS OF THE PROJECT

****

1. **Image taking Process**

****

1. **Recognizing face and expression Process**

# CHAPTER -8 CONCLUSION

Facial expression plays an important role in communication and thus identifying the correct expression is as essential as knowing the exact matter of the communication. This project proposes an approach for recognizing the category of facial expressions. Face Detection and Extraction of expressions from facial images has been achieved successfully and is useful in many applications, such as robotics vision, video surveillance, digital cameras, security and human-computer interaction. This project’s objective is to develop a FACE RECOGNITION AND EMOTION RECOGNITION implementing the computer visions and enhancing the advanced feature extraction and classification in face expression recognition.

In this project, seven different facial expressions of different persons’ images from different datasets have been analyzed. This project involves facial expression pre-processing of captured facial images followed by feature extraction using feature extraction using Local Binary Patterns and classification of facial expressions based on training of datasets of facial images based on Support Vector Machines. This project recognizes more facial expressions based on Haarcascade and fer2013 face database.

# CHAPTER-9 FUTURE REPLACEMENT

In future we can add features like:

1. Registering person details automatically when camera detects a human face.
2. Registering multiple person’s details simultaneously.
3. Doctors can use the system to understand the intensity of pain or illness of a deaf patient.

# CHAPTER-10 BIBLIOGRAPHY

* “Facial Emotion Analysis using Deep Convolution Neural Network”, Rajesh Kumar G A, Ravi Kant Kumar, Goutam Sanyal, IEEE, 2017.
* "Rapid object detection using a boosted cascade of simple features.”, Viola, Paul, and Michael Jones. Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Vol. 1. IEEE, 2001
* “Understanding of a Convolutional Neural Network”, Saad ALBAWI, Tareq Abed MOHAMMED, Saad AL-ZAWI, ICET2017.
* [https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade\_](https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_%20eye.xml) [eye.xml](https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_%20eye.xml)

<https://www.youtube.com/watch?v=Dqa-3N8VZbw&t=431s>