A PROJECT REPORT

ON HUMAN EMOTION ANALYSIS USING DEEP LEARNING TECHNIQUES.

Submitted by

BURUJU SOWMYA - 18STUCHH010005

Under Guidance of

Dr. Shubhangi urkude .V

A Final Project report submitted to the Department of computer Science for partial fulfilment of academic requirement for the Special Project course

Department of Computer Science & Technology
The ICFAI Foundation for Higher Education
Faculty of Science & Technology
(a Deemed University under Section 3 of UGC Act. 1956)
Donthanapally, Shankarapalli Road, Hyderabad – 501203

September – December, 2020

TABLE OF CONTENTS

1.Abstract 5

2.Introduction 6

3.Literature survey 8

4.Requirements Specification 11

4.1 Software Requirements 11

4.2 Hardware Requirements 11

5.Software Specifications 11

5.1 Python 11

5.2Anaconda 12

5.3 Pycharm 12

6.Planning 13

7.Design 14

8.Opencv-python 18

9.Videocapture 19

10.Videowriter 19

11.Haarcascade 20

12.Classification 20

12.1 Training phase 21

12.2 Testing phase 21

13.Code 22

14.Output 24

15.Proposed method 27

16.Dataset 28

16.1 Face detection 29

16.2 Feature extraction 29

16.3 Face feature extraction 30

16.4 Python pipeline 34

16.5 Machine learning 34

17.Future scope 36

18.Conclusion 35

19.References 37

LIST OF TABLES AND FIGURES

Table 1: Emotion recognition different approach and successes 11

Fig 1 : planning model 14

Fig 2: DFD level 0 16

Fig 3: DFD level 1 16

Fig 4: DFD level 2 face detection 17

Fig 5: Emotion classification 17

Fig 6: execution output of face recognition 24

Fig 6.1: execution output of eyes recognition 25

Fig 6.2: execution output of smile recognition 26

Figure 7. Cohn-Kanade CK+ database (above) have frontal facial images with stable illuminance. The Facial Expression Recognition 2013 (FER-2013) Dataset (below) has images cropped from movies varies in head posture and illuminance 28

Table 2 : Feature extraction algorithms 30

Fig 8: performance analysis for Weighted PCA approach 31

Fig 9: Performance analysis of gavor Wavelet Method 31

Fig 10: Haar-Like Feature 32

Fig 11: The classifer cascade is a chain of filters .Image subregions that make it through the entire cascade are classified as face all others not face 33

Fig 12: Detection of face Features Such as Eyes Nose and Lips 33

ABSTRACT

People share a general and significant plan of emotions which are shown through dependable outward appearances.Most of the time human sentiments are extensively spoken with eye and mouth articulation.Introduced here is a mutt incorporate extraction and outward appearance affirmation method that utilizes Viola-Jones course object locaters and Harris corner key-centers to isolate faces and facial features from pictures and uses head fragment investigation,direct discriminant assessment, histogram-of-arranged points (Hoard) incorporate extraction, and support vector machines (SVM) to set up a multi-class marker for masterminding the seven essential human outward appearances. The important module realizes skin acknowledgment, using Markov self-assertive field models for picture division and skin disclosure. A ton of a couple of concealed pictures with human appearances have been considered as the planning set. An ensuing module is at risk for eye and mouth revelation and extraction. The specific module uses the HLV concealing space of the foreordained eye and mouth area. The third module recognizes the sentiments envisioned in the eyes and mouth, using edge area and assessing the incline of eyes and mouth's region figure.

INTRODUCTION

Facial feeling recognition is the way toward recognizing human feelings from outward appearances. The human mind perceives feelings consequently, and programming has now been built up that can perceive feelings also. This innovation is turning out to be more exact constantly, and will in the end have the option to peruse feelings just as our minds do.

Artificial intelligence can identify feelings by realizing what every outward appearance implies and applying that information to the new data introduced to it. Enthusiastic man-made consciousness, or feeling AI, is an innovation that is fit for perusing, mimicking, deciphering, and reacting to human outward appearances and feelings.

Most common exposition of an idea of emotion could be found as "a natural instinctive state of mind deriving from one's circumstances, mood, or relationships with others". Which misses depicting the driving force behind all motivation which may positive, negative or neutral. This is very important information to understand emotion as an intelligent agent. It is very complicated to detect the emotions and distinguish among them. Before a decades or two emotion started to become a concern as an important addition towards the modern technology world. Rises the hope of new dawn for intelligence apparatus. Imagine a world where machines do feel what humans need or want. With the special kind of calculation then that machine could predict the further consequences and by which mankind could avoid serious circumstances and lot more. Humans are far more strong and intelligent due to the addition of the emotion but less effective than machines. But what if machines get this special features of human? It will be the strongest addition to the technology ever. And to make the dreams come true this is the first step; train a system to spot and recognize emotions. This is the start of an intelligent system. Intelligent

Systems are becoming more efficient by predicting and classifying decision in various aspects of practical life. Particularly emotion recognition through deep learning has become an intriguing research area for its innovative nature and practical implication. This technique mainly consists of detecting emotion through various kinds of input taken from different human behavior and condition. A technology namely neural network detects emotion through deep learning. For its complication mentioned earlier, an emotion recognition system with stellar efficiency and accuracy is needed.

Human emotions can be classified as: fear, contempt, disgust, anger, surprise, sad, happy, and neutral. These emotions are very subtle. Facial muscle contortions are very minimal and detecting these differences can be very challenging as even a small difference results in different expressions . Also, expressions of different or even the same people might vary for the same emotion, as emotions are hugely context dependent . While we can focus on only those areas of the face which display a maximum of emotions like around the mouth and eyes , how we 2 extract these gestures and categorize them is still an important question. Neural networks and machine learning have been used for these tasks and have obtained good results. Machine learning algorithms have proven to be very useful in pattern recognition and classification. The most important aspects for any machine learning algorithm are the features. In this paper we will see how the features are extracted and modified for algorithms like Support Vector Machines .

We will compare algorithms and the feature extraction techniques from different papers. The human emotion dataset can be a very good example to study the robustness and nature of classification algorithms and how they perform for different types of dataset. Usually before extraction of features for emotion detection, face detection algorithms are applied on the image or the captured frame. We can generalize the emotion detection steps as follows:

1) Dataset preprocessing
2) Face detection
3) Feature extraction
4) Classification based on the features

In this work, we focus on the feature extraction technique and emotion detection based on the extracted features. Section 2 focuses on some important features related to the face. Section 3 gives information on the related work done in this field. Related work covers many of the feature extraction techniques used until now. It also covers some important algorithms which can be used for emotion detection in human faces. Section 4 details the tools and libraries used in the implementation. Section 5 explains the implementation of the proposed feature extraction and emotion detection framework. Section 6 highlights the result of the experiment. Section 7 covers the conclusion and future work.

Literature Survey

Previous works are focused on eliciting results from unimodal systems. Machines used to predict emotion by only facial expressions or only vocal sounds. After a while, multimodal systems that use more than one feature to predict emotion are more effective and give more accurate results. So that, the combination of features such as audio-visual expressions, EEG, body gestures have been used since. More than one intelligent machine and neural networks are used to implement the emotion recognition system.Research has demonstrated that deep neural networks can effectively generate discriminative features that approximate the complex non-linear dependencies between features in the original set.

Yelin Kim and Emily Mower Provost explore whether a subset of an utterance can be used for emotion inference and how the subset varies by classes of emotion and modalities. They propose a windowing method that identifies window configurations, window duration, and timing, for aggregating segment-level information for utterance-level emotion inference. The experimental results using the IEMOCAP and MSPIMPROV datasets show that the identified temporal window configurations demonstrate consistent patterns across speakers, specific to different classes of emotion and modalities. They compare their proposed windowing method to a baseline method that randomly selects window configurations and a traditional all-mean method that uses the full information within an utterance. This method shows a significantly higher performance in emotion recognition while the method only uses 40–80% of information within each utterance. The identified windows also show consistency across speakers, demonstrating how multimodal cues reveal emotion over time. These patterns also align with psychological findings. But after all achievement, the result is not consistent with this method.

Anagha S. Dhavalikar has proposed Automatic Facial Expression recognition system. In This system there are three phase 1. Face detection 2. Feature Extraction and 3. Expression recognition. The First Phase Face Detection are done by RGB Color model, lighting compensation for getting face and morphological operations for retaining required face i.e. eyes and mouth of the face. This System is also used AAM i.e. Active Appearance Model Method for facial feature extraction. In this method the point on the face like eye, eyebrows and mouth are located and it create a data file which gives information about model points detected and detect the face an expression is given as input AAM Model changes according to expression
Yong-Hwan Lee, Woori Han and Youngseop Kim have proposed system based on Bezier curve fitting. This system used two steps for facial expression and emotion first one is detection and analysis of facial area from input original image and next phase is verification of facial emotion of characteristics feature in the region of interest. The first phase for face detection it uses color still image based on skin color pixel by initialized spatial filtering ,based on result of lighting compassion then to estimate face position and facial location of eye and mouth it used feature map After extracting region of interest this system extract points of the feature map to apply Bezier curve on eye and mouth The for understanding of emotion this system uses training and measuring the difference of Hausdorff distance With Bezier curve between entered face image and image from database

Zixing Zhang, Fabien Ringeval, Eduardo Coutinho, Erik Marchi and Björn Schüller proposed some improvement in SSL technique to improve the low performance of a classifier that can deliver on challenging recognition tasks reduces the trust ability of the automatically labeled data and gave solutions regarding the noise accumulation problem - instances that are misclassified by the system are still used to train it in future iterations . they exploited the complementarity between audio-visual features to improve the performance of the classifier during the supervised phase. Then, they iteratively re-evaluated the automatically labeled instances to correct possibly mislabeled data and this enhances the overall confidence of the system's predictions. This technique gives the best possible performance using SSL technique where labeled data is scarce and/or expensive to obtain but still, there are various inherent limitations that limit its performance in practical applications. This technique has been tested on a specific database with a limited type and number of data. The algorithm which has been used is not capable of processing physiological data alongside other types of data.

Y. Fan, X. Lu, D. Li, and Y. Liu. proposed a method for video-based emotion recognition in the wild. They used CNN-LSTM and C3D networks to simultaneously model video appearances and motions [16]. They found that the combination of the two kinds of networks can give impressive results, which demonstrated the effectiveness of the method. In their proposed method they used LSTM (Long Short Term Memory) - a special kind of RNN, C3D – A Direct Spatio-Temporal Model and Hybrid CNN-RNN and C3D Networks. This method gives great accuracy and performance is remarkable. But this method is much convoluted, time-consuming and less realistic. For this reason, efficiency is not that impressive

.

Reference and year	Approach and Method	Performance
Wei-Long Zheng and BaoLiang Lu (2016)	EEG-based affective models without labeled target data using transfer learning techniques (TCA-based Subject Transfer)	Positive (85.01%) emotion recognition rate is higher than other approaches but neutral (25.76%) and negative (10.24%) emotions are often confused with each other.
Zixing Zhang, Fabien Ringeval, Fabien Ringeval, Eduardo Coutinho, Erik Marchi and Björn Schüller (2016)	Semi-Supervised Learning (SSL) technique	Delivers a strong performance in the classification of high/low emotional arousal (UAR = 76.5%), and significantly outperforms traditional SSL methods by at least 5.0% (absolute gain).

Y. Fan, X. Lu, D. Li, and Y. Liu. (2016)	Video-based Emotion Recognition Using CNN-RNN and C3D Hybrid Networks	Achieved accuracy 59.02% (without using any additional Emotion labeled video clips in training set) which is the best till now.
A. Yao, D. Cai, P. Hu, S. Wang, L. Shan and Y. Chen (2016)	HoloNet: towards robust emotion recognition in the wild	Achieved mean recognition rate of 57.84%.
Yelin Kim and Emily Mower Provos (2016)	Data driven framework to explore patterns (timings and durations) of emotion evidence, specific to individual emotion classes	Achieved 65.60% UW accuracy, 1.90% higher than the baseline.

Table 1: Emotion recognition different approach and successes

Requirements Specification

1.Software Requirements

• Language : Python, HTML-5, CSS, JavaScript

• Operating System : Windows 8.1 or Above

• Libraries : OpenCV, EEL, NumPy, FisherFace

2. Hardware Requirements

• RAM Capacity : 4GB

• Memory : 150 MB

• Graphics Card : 1 GB

• Accessories : Laptop or a Computer with a webcam

Software Specifications

1.Python

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). Python allows programming in ObjectOriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech- giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc. The biggest strength of the Python is large library.

Python is a powerful scripting language and is very useful for solving statistical problems involving machine learning algorithms. It has various utility functions which help in preprocessing. Processing is fast and it is supported on almost all platforms. Integration with C++ and other image libraries is very easy, and it has in-built functions and libraries to store and manipulate data of all types. It provides the pandas and numpy framework which helps in manipulation of data as per our need. A good feature set can be created using the numpy arrays which can have n-dimensional data.

2.Anaconda

Anaconda is free and open source distribution of the Python and R programming languages for Scientific Computing (data science, machine learning applications, large-scale data processing, Predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the Package Management System Conda. The Anaconda distribution is used by over 15 million users and includes more than 1500 popular data-science packages suitable for Windows, Linux, and MacOS

3.Pycharm

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains.[6] It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda.[7]PyCharm is cross-platform,with Windows, macOS and Linux versions. The Community Edition is released under the Apache License,[8] and there is also Professional Edition with extra features – released under a proprietary license.

Planning

The steps we followed while developing this project are-:

1. Analysis of the problem statement

2. Gathering of the requirement specification

3. Analysation of the feasibility of the project.

4. Development of a general layout.

5. Going by the journals regarding the previous related works on this field.

6. Choosing the method for developing the algorithm.

7. Analyzing the various pros and cons.

8. Starting the development of the project

9. Installation of software like ANACONDA.

10. Developing an algorithm.

11. Analysation of algorithm by guide.

12. Coding as per the developed algorithm in PYTHON.

We developed this project as per the iterative waterfall model:

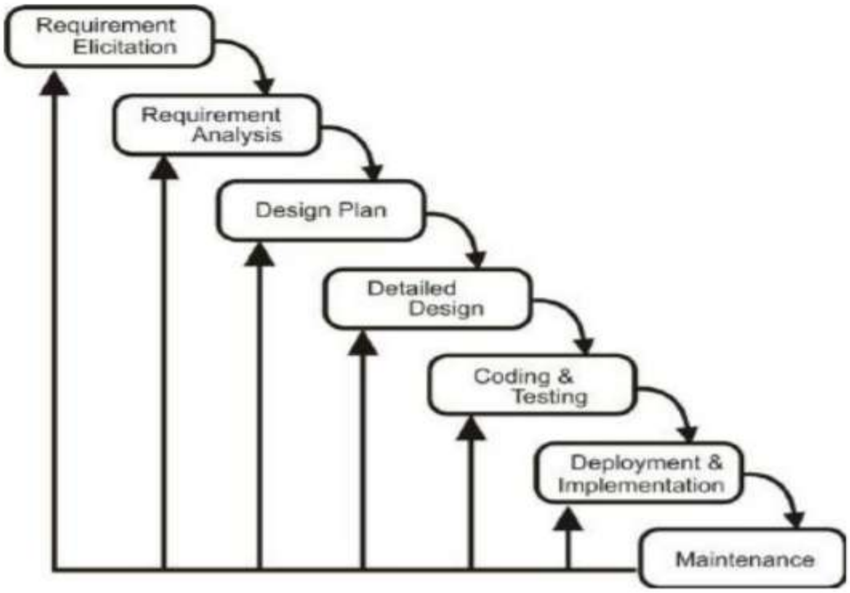


Fig 1 : planning model

DESIGN

DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about process timing or whether processes will operate in sequence or in parallel, unlike a traditional structured flowchart which focuses on control flow, or a UML activity workflow diagram, which presents both control and data flows as a unified model.

Data flow diagrams are also known as bubble charts. DFD is a designing tool used in the topdown approach to Systems Design.

Symbols and Notations Used in DFDs

Using any convention‟s DFD rules or guidelines, the symbols depict the four components of data flow diagrams .

External entity: an outside system that sends or receives data, communicating with the system being diagrammed. They are the sources and destinations of information entering or leaving the system. They might be an outside organization or person, a computer system or a business system. They are also known as terminators, sources and sinks or actors. They are typically drawn on the edges of the diagram.

Process: any process that changes the data, producing an output. It might perform computations, or sort data based on logic, or direct the data flow based on business rules.

Data store: files or repositories that hold information for later use, such as a database table or a membership form.

Data flow: the route that data takes between the external entities, processes and data stores. It portrays the interface between the other components and is shown with arrows, typically labeled with a short data name, like “Billing details.”

DFD levels and layers

A data flow diagram can dive into progressively more detail by using levels and layers, zeroing in on a particular piece. DFD levels are numbered 0, 1 or 2, and occasionally go to even Level 3 or beyond. The necessary level of detail depends on the scope of what you are trying to accomplish.

DFD Level 0 is also called a Context Diagram. It‟s a basic overview of the whole system or process being analyzed or modeled. It‟s designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its subprocesses.

DFD Level 2 then goes one step deeper into parts of Level 1. It may require more text to reach the necessary level of detail about the system‟s functioning.

Progression to Levels 3, 4 and beyond is possible, but going beyond Level 3 is uncommon. Doing so can create complexity that makes it difficult to communicate, compare or model effectively.

Using DFD layers, the cascading levels can be nested directly in the diagram, providing a cleaner look with easy access to the deeper dive.

Level 0

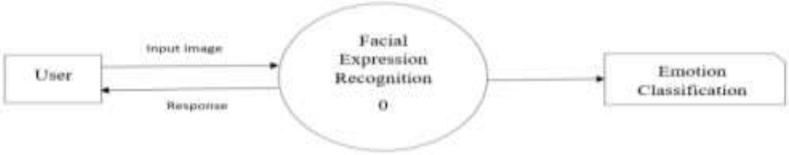


Fig 2: DFD level 0

Level 1

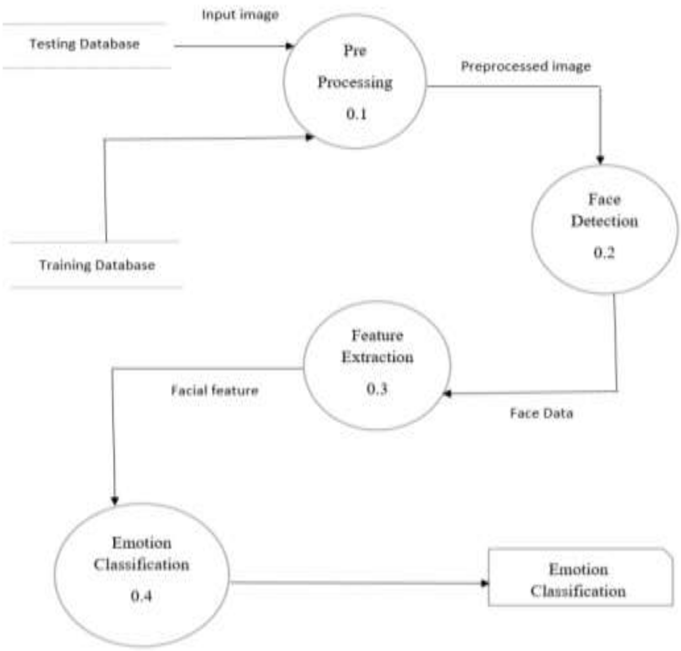


Fig 3: DFD level 1

Level 2

Face Detection-

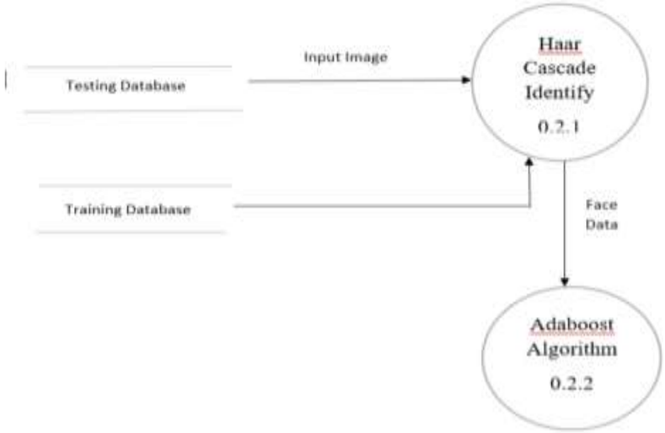


Fig 4: DFD level 2 face detection.

Emotion Classification

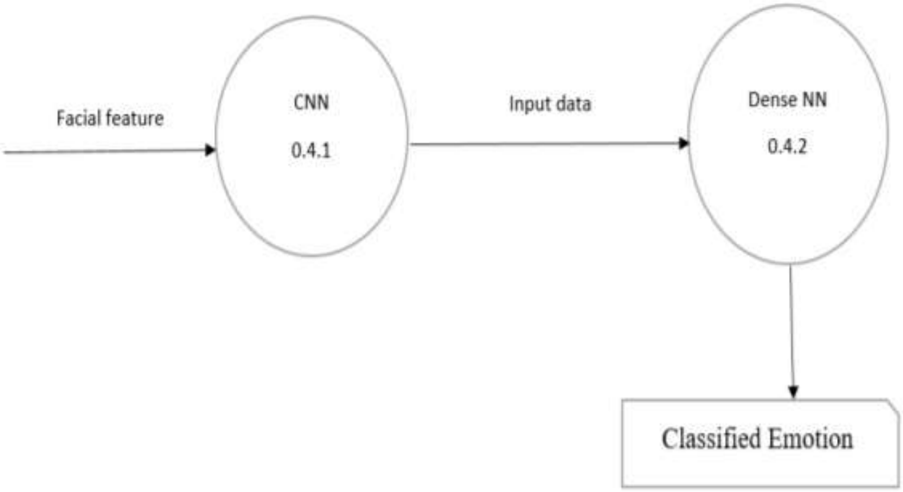


Fig 5: Emotion classification.

OPENCV-PYTHON

OpenCV was begun at Intel in 1999 by Gary Bradsky and the primary delivery turned out in 2000. Vadim Pisarevsky joined Gary Bradsky to deal with Intel's Russian programming OpenCV group. In 2005, OpenCV was utilized on Stanley, the vehicle who won 2005 DARPA Grand Challenge. Later its dynamic improvement proceeded under the help of Willow Garage, with Gary Bradsky and Vadim Pisarevsky driving the undertaking. At this moment, OpenCV underpins a great deal of calculations identified with Computer Vision and Machine Learning and it is extending step by step.

At present OpenCV upholds a wide assortment of programming dialects like C++, Python, Java and so on and is accessible on various stages including Windows, Linux, OS X, Android, iOS and so forth Additionally, interfaces dependent on CUDA and OpenCL are likewise under dynamic advancement for fast GPU activities.

OpenCV-Python is the Python API of OpenCV. It joins the best characteristics of OpenCV C++ API and Python language.Around 3000 calculations are at present implanted inside OpenCV library, every one of these calculations being proficiently enhanced. It underpins ongoing vision applications. These calculations are classified under exemplary calculations, condition of craftsmanship PC vision calculations and AI calculations. These calculations are handily actualized in Java, MATLAB, Python, C, C++ and so forth and are all around upheld by working framework like Windows, Mac OS, Linux and Android.

The library has more than 2500 calculations. These can be utilized to distinguish and perceive faces, recognize objects, arrange human activities in recordings, track camera developments, track moving items, extricate 3D models of objects,etc. Alongside settled organizations like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that utilize the library, there are numerous new companies, for example, Applied Minds, VideoSurf, and Zeitera, that utilize OpenCV

VideoCapture:

Python gives different libraries to picture and video preparing. One of them is OpenCV. OpenCV is a tremendous library that helps in giving different capacities to picture and video tasks. With OpenCV, we can catch a video from the camera. It lets you make a video catch object which is useful to catch recordings through webcam and afterward you may perform wanted procedure on that video.

Steps to catch a video:

- Use cv2.VideoCapture() to get a video catch object for the camera.
- Set up a boundless while circle and utilize the read() technique to peruse the casings utilizing the above made item.
- Use cv2.imshow() technique to show the casings in the video.
- Breaks the circle when the client clicks a particular key.

VideoWriter:

For saving images, we use cv2.imwrite() which saves the image to a specified file location. But, for saving a recorded video, we create a Video Writer object (cv2.VideoWriter(filename,fourcc,fps,frameSize)) OpenCV is an open-source and most famous PC vision library that contains a few PC vision calculations. You can peruse, show, compose and doing bunches of different procedure on pictures and recordings utilizing OpenCV. The OpenCV module by and large utilized in famous programming dialects like C++, Python, Java. To spare a video in OpenCV cv.VideoWriter() strategy is utilized.

Syntax: cv.VideoWriter(filename, fourcc, fps, frameSize)
Parameters:
filename: Input video file
fourcc: 4-character code of codec used to compress the frames
fps: framerate of videostream
framesize: Height and width of frame

HAARCASCADE

Object Detection is a PC innovation identified with PC vision, picture handling and profound discovering that manages distinguishing examples of objects in pictures and recordings. We will do object detection in this article utilizing something known as haar falls.

Haar Cascade classifiers are a successful path for object identification. This strategy was proposed by Paul Viola and Michael Jones in their paper Rapid Object Detection utilizing a Boosted Cascade of Simple Features .Haar Cascade is an AI based methodology where a great deal of positive and negative pictures are utilized to prepare the classifier.

- Positive pictures – These pictures contain the pictures which we need our classifier to recognize.
- Negative Images – Images of everything else, which don't contain the article we need to identify.

Requirements

- Ensure you have python, Matplotlib and OpenCV introduced on your pc (all the most recent variants).
- The haar course records can be downloaded from the OpenCV Github store.

Classification

Classification is a general process related to categorization. It is the action or process of clustering something. System is capable to classify images into different emotions. For classification of images, system will use Support Vector Machine (SVM). There are two phases in preparing an efficient classifier.

Training Phase:

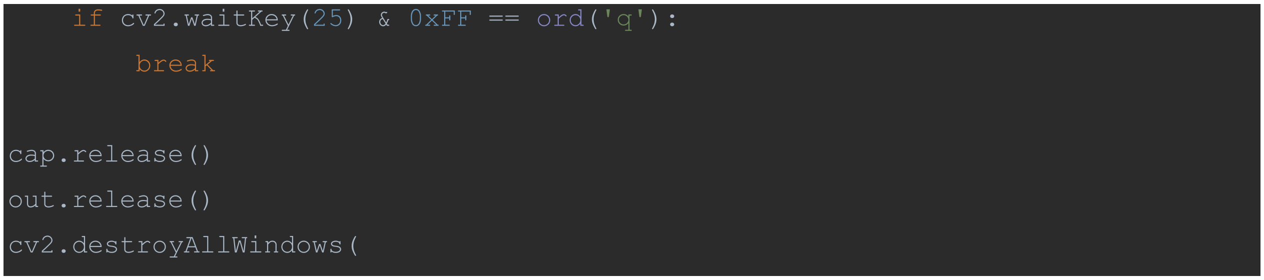
The training dataset is divided into four classes for three moods i.e. happy, neutral, sad, and angry for each of these corresponding mood values +1, +2, +3 and +4 is assigned in the training file as the correct label for this model to train. Here giving feature point along with its correct label is at most important as the learning of the model is supervised learning. The feature point is the position of the landmark on the face which were calculated in the previous step of feature extraction these feature points are converted in a single dimension from a two-dimension X, Y coordinate system.

Testing Phase:

In testing phase, a similar input is given to the train model as it was in training phase with an exception that correct label is not known by the system for predicting the mood. It is just use to check the accuracy of the predicted result. It processes image from IFEE database, detecting the face using the HOG technique, then detecting the facial landmark points and classifying it using SVM classifier to predict the mood happy.

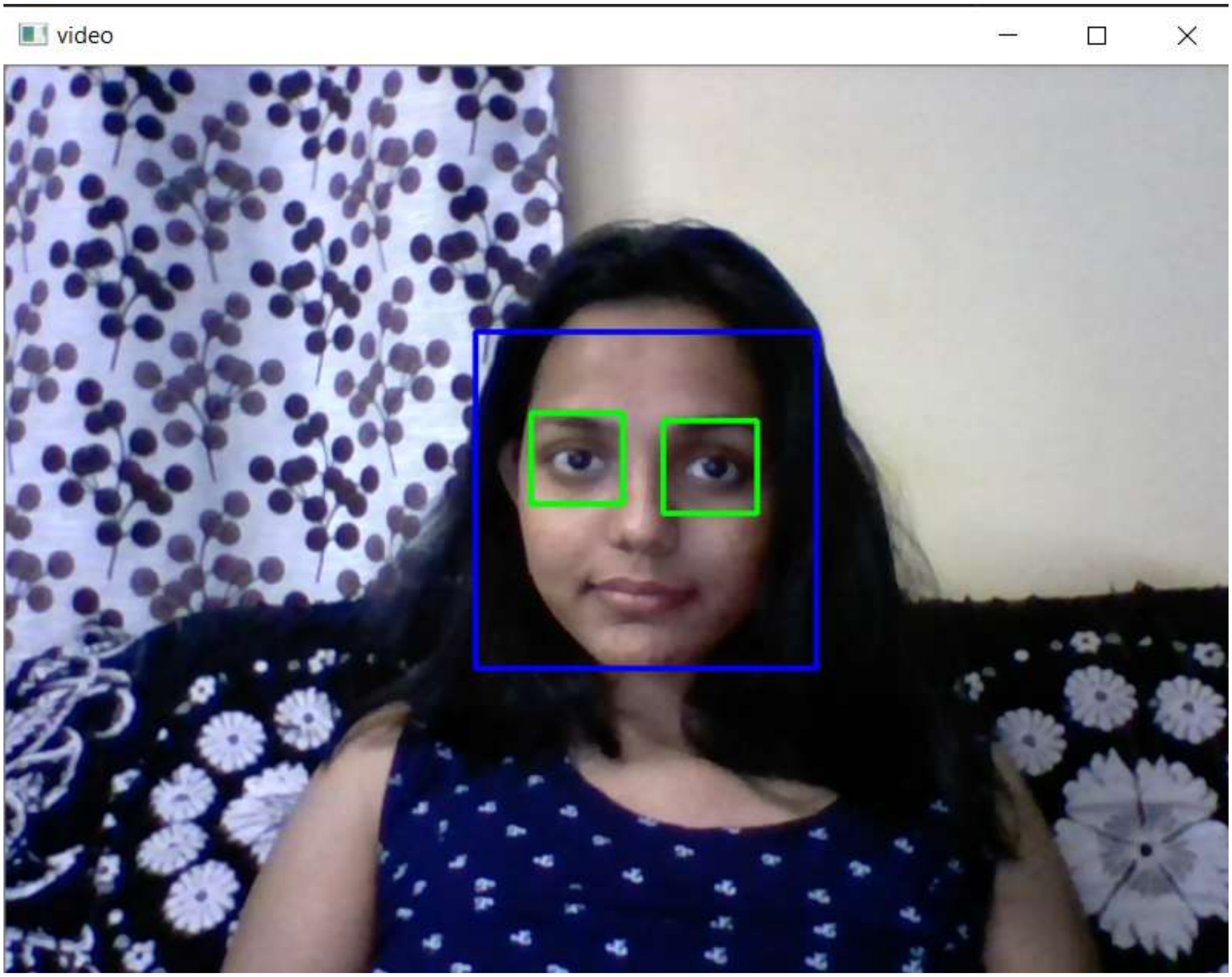
Code:

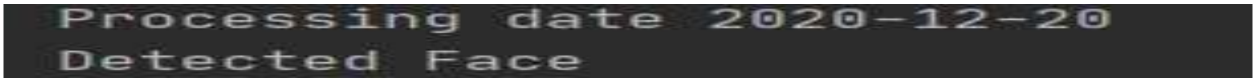
```
import cv2

from datetime import datetime

full_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_fullbody.xml')
eye_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_eye.xml')
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')
smile_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_smile.xml')

cap = cv2.VideoCapture(0)
ret, Frame = cap.read()
fourcc = cv2.VideoWriter_fourcc(*'XVID')

# name as date + hour

current_date = datetime.today().strftime('%Y-%m-%d')
filename = datetime.today().strftime('%Y-%m-%d') + ".avi"
print(filename)
out = cv2.VideoWriter(filename, fourcc, 20.0, (640, 480))

while 1:
    processing_date = datetime.today().strftime('%Y-%m-%d')
    print("Processing date " + processing_date)

    if processing_date != current_date:
        out.release()
        current_date = datetime.today().strftime('%Y-%m-%d')
        filename = datetime.today().strftime('%Y-%m-%d') + ".avi"
        # start new Video writer with new date
        out = cv2.VideoWriter(filename, fourcc, 20.0, (640, 480))
    ret, Frame = cap.read()
    gray = cv2.cvtColor(Frame, cv2.COLOR_BGR2GRAY)
    human = full_cascade.detectMultiScale(gray, 1.1, 4)

    for (x, y, w, h) in human:
        cv2.rectangle(Frame, (x, y), (x + w, y + h), (0, 0, 220), 3)

    faces = face_cascade.detectMultiScale(gray, 1.3, 5)

    for (x, y, w, h) in faces:
        cv2.rectangle(Frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
        print("Detected Face")
        roi_gray = gray[y:y + h, x:x + w]
        roi_color = Frame[y:y + h, x:x + w]

        eyes = eye_cascade.detectMultiScale(roi_gray)
        # out.write(Frame)

        for (ex, ey, ew, eh) in eyes:
            cv2.rectangle(roi_color, (ex, ey), (ex + ew, ey + eh), (0, 255,
0), 2)
        print("Detected Eyes")
        out.write(Frame)

        smiles = smile_cascade.detectMultiScale(roi_gray, 1.8, 20)
        for (sx, sy, sw, sh) in smiles:
            cv2.rectangle(roi_color, (sx, sy), ((sx + sw), (sy + sh)), (0, 0,
255), 2)
        print("Detected Smile")
        out.write(Frame)

    cv2.imshow('video', Frame)
    # out.write(Frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```



```
    if cv2.waitKey(25) & 0xFF == ord('q'):
        break

cap.release()
out.release()
cv2.destroyAllWindows(
```

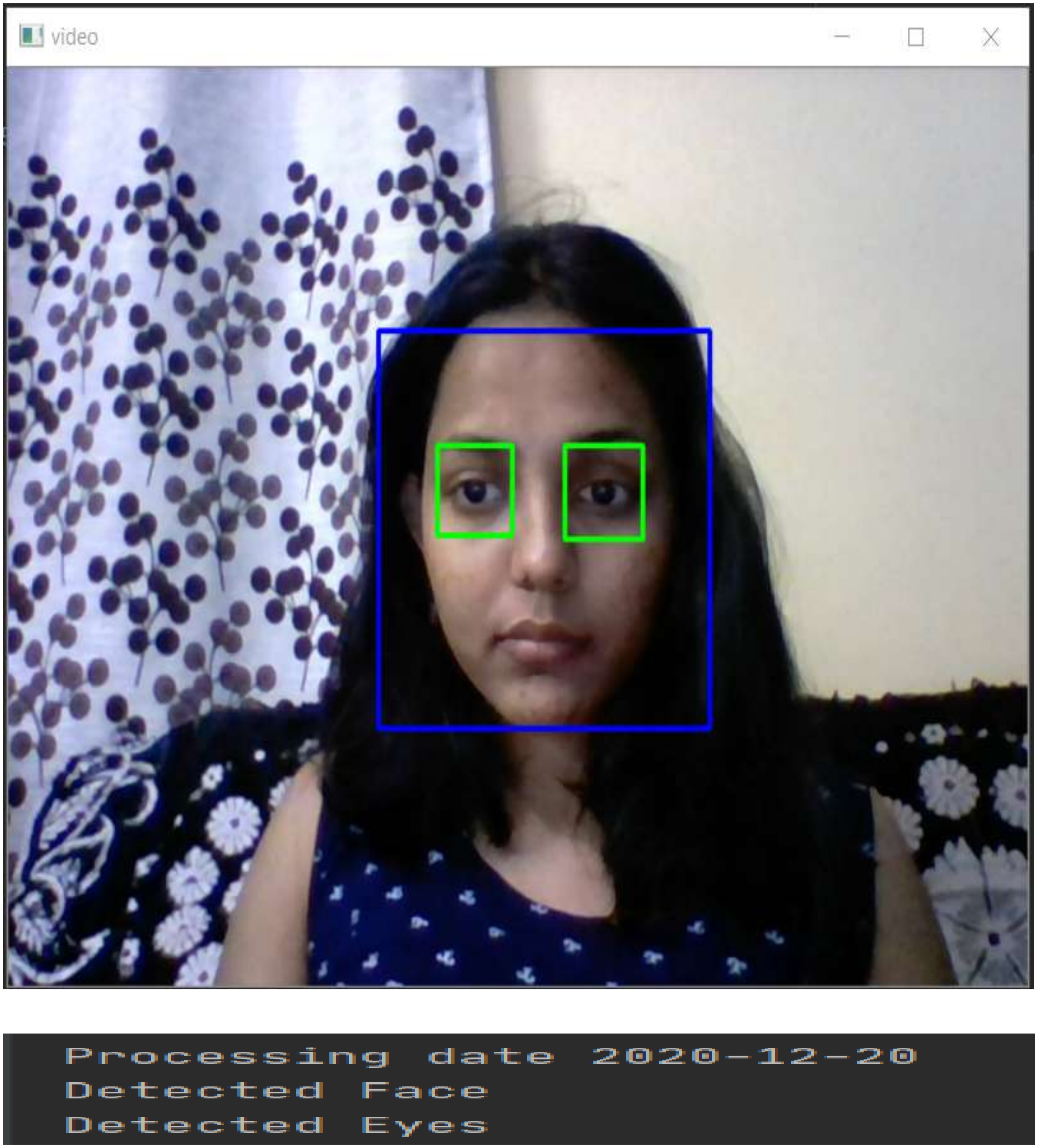
Output:





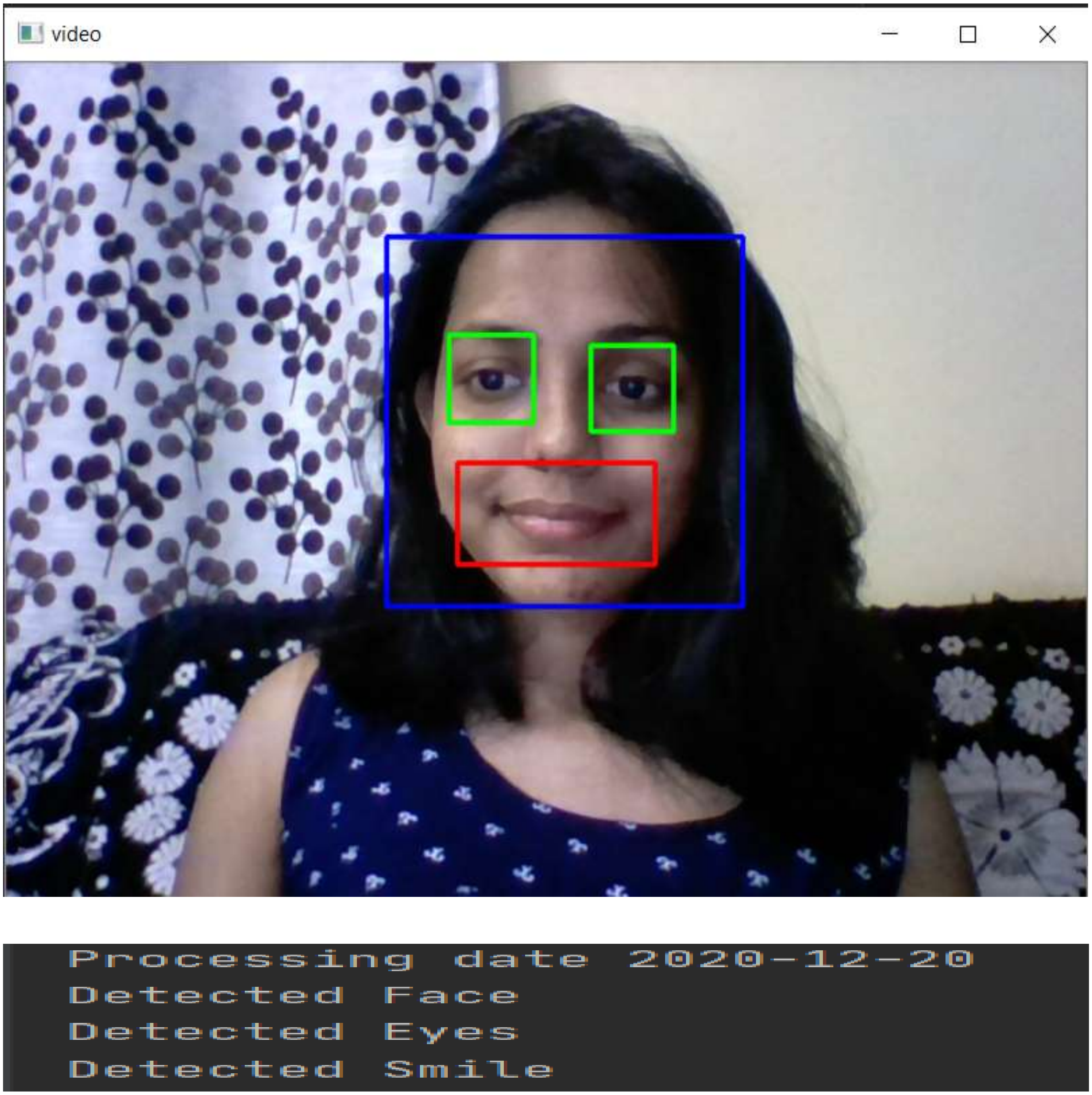
```
  Processing date 2020-12-20
  Detected Face
```

Fig 6: execution output of face recognition



```
Processing date 2020-12-20
Detected Face
Detected Eyes
```

Fig 6.1: execution output of eyes detection



```
Processing date 2020-12-20
Detected Face
Detected Eyes
Detected Smile
```

Fig 6.2: execution output of smile detection

Proposed Method

The study of automatic human facial emotion recognition started from defining and categorizing human facial expressions. After that, researchers built databases that contained labelled facial expression examples. Finally, various approaches have been used to recognize human emotions.

The study of facial emotion recognition can be traced back to 1970s. Paul Ekman and his colleagues found that there are six facial expressions (happy, sadness, anger, fear, surprise, disgust) can be understood by people from different cultures. The difference in backgrounds influences facial expressions mainly in intensity. For example, watching the same comedy film, Americans tend to laugh with their mouth widely open while Japanese are more 1 likely to smile without showing their teeth. The observation that infants are able to show a wide range of facial expressions and respond to facial expressions from others without being taught suggesting that the ability to deliver emotions and understand emotions via facial expressions is inherent in humans.

Emotion recognition can have interesting applications in human-robot interaction, thus paving the way for a scenario where human-robot interaction will normally take place in the real world. When a speaker expresses an emotion while adhering to an inconspicuous intonation pattern, human listeners can nevertheless perceive the emotional information through the pitch and intensity of speech. On the other hand, our aim is to capture the diverse acoustic cues that are in the speech signal and to analyze their mutual relationship to the speaker’s emotion. We propose a technique to recognize several basic emotions, namely sadness (SAD), anger (ANG), surprise (SUR), fear (FEA), happiness (HAP) and disgust (DIS), based on the analysis of phonetic and acoustic properties.

Data set

A few informational indexes have been set up to construct feeling acknowledgment models and assess exhibitions. Same techniques can get drastically various outcomes on various informational indexes. This is because of the fluctuation in informational collections. Facial feeling information bases can be isolated into two classifications: lab informational indexes like CohnKanade CK+ database[16] and wild informational indexes like Acted Facial Expressions in the Wild (AFEW) [5]. In last mentioned, outward appearances pictures or recordings are browsed films and online recordings which different altogether in goal, light, head present and so on while most lab informational indexes control every one of these variables cautiously.

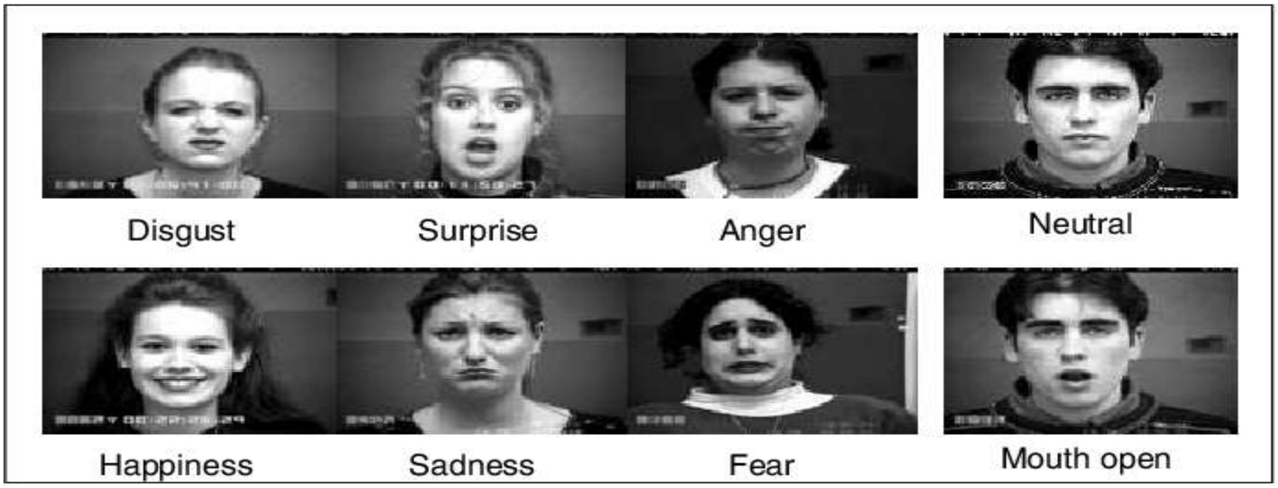


Figure 7. Cohn-Kanade CK+ database (above) have frontal facial images with stable illuminance. The Facial Expression Recognition 2013 (FER-2013) Dataset (below) has images cropped from movies varies in head posture and illuminance.

In terms of emotion recognition, there is no indefinite way or method which is the univocal solution. A lot of solution have come and many to comes in near future with significant improvement in terms of efficiency, accuracy, and usability. In past and the current research shows that multimodalities dominated the area of emotion recognition than unimodality. Using EEG and audio-visual signal yields the best possible results according to the newest researches. We assume LSTMRNN is the best way to handle multimodalities. So our proposal is focused on emotion recognition by EEG and audio-visual signal using LSTM-RNN. This type of research has been done before. But our challenge is to improve the model where it will be trained by EEG and audiovisual data at the same time and will make a relation between this data wherein, if one type of data is not available in a situation, the model could still produce the result; finding the relation within the data. So, the training will have two part; training for the data and training to understand the relations between the data.

1.Face Detection

It is important for an image to be well lit and recognizable for the system to detect face. The system should read faces in low light but if the light is below certain accepted limit the system gives a prompt to retake as no face is detected. Image is divided into the fixed size blocks. The blocks from which Histogram of Gradient (HOG) features are extracted and are overlapped on the detector window, the extracted vectors are fed as input to linear SVM in order to classify object or non-object classification and obtain result as the extracted face.

Face detection was the first and important part of the processing pipeline. Before further processing, we had to detect the face, even though our images contained only frontal facial expression data. Once the face was detected, it was easier to determine the region of interest and extract features from it.

For face detection, we tried many algorithms like Haar-cascades from OpenCV. Finally we settled for face detector based on histogram of oriented gradients from Dlib library. HoG 27 descriptors along with SVM are used to identify the face from the image. Images are converted to grayscale and resized for uniformity.

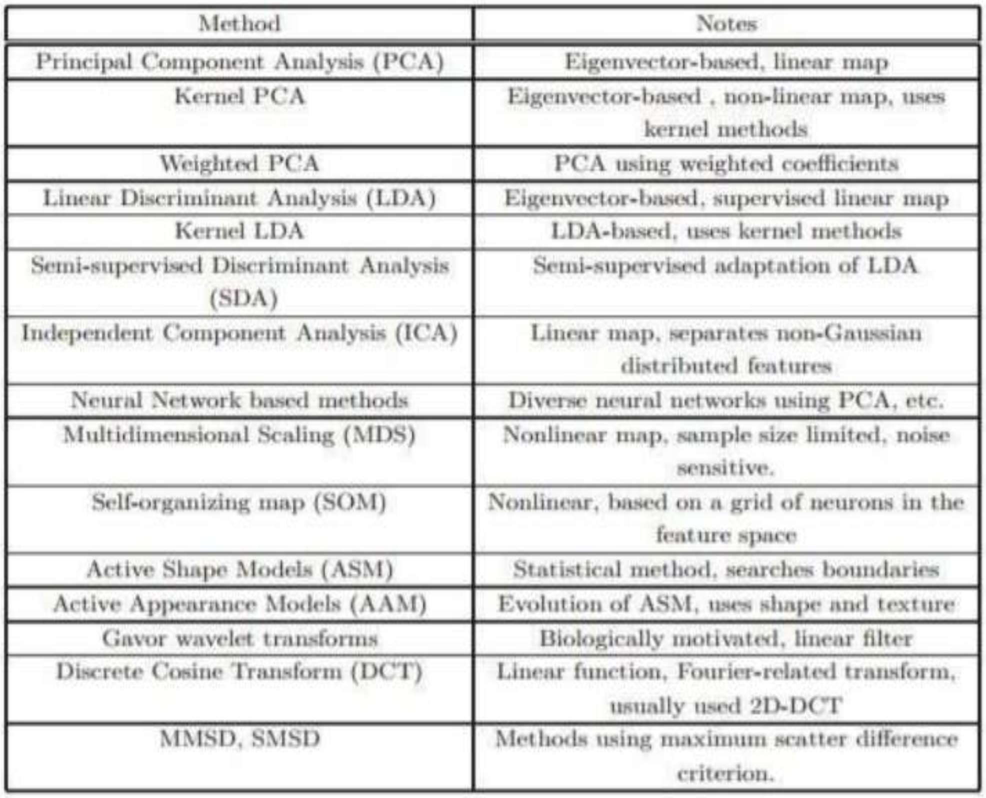
2. Feature Extraction

When the system takes the images as an input it’s very important to device an algorithm to detect emotions. This detection of emotions can be obtained by extracting the feature of the face for example the lips are stretched and make an extended ‘v’ type structure it means person is very happy similarly if eyebrows are arched and lips are in the shape of small ‘n’ then it amounts to a sad emotions. Number of permutations and combinations like the above stated determine the appropriate emotion through face and classify them into different present emotion categories.

Feature extraction is necessary stage for predicting the correct mood of the user so proposed system will use ensemble of regression trees to obtain facial landmark positions from sparse subset of pixel intensities. For robust performance with high quality prediction gradient boosting for training an ensemble of regression trees is used to minimize the misclassified and partially labelled data. There are total 52 landmark points on the face that system will detect and these points will be the data on basis of which the mood will be predicted . These extracted feature points are further passed to the SVM for classification.

3.Face Feature Extraction

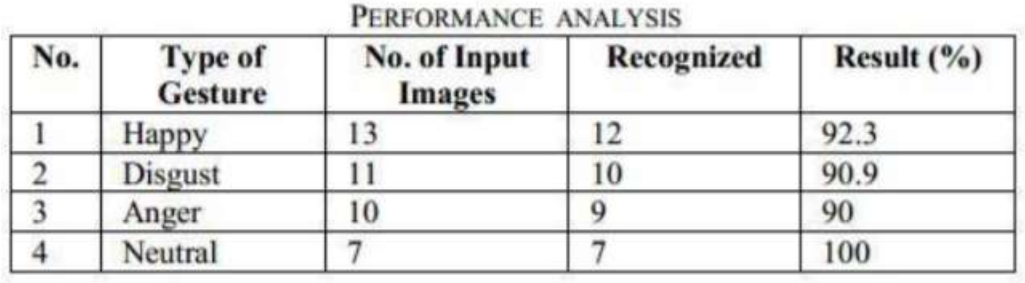
Now for Feature extraction there are many methods to apply. They can be categorized according to whether they focus on motion or deformation of faces and facial features, respectively, whether they act locally or holistically. In our case we need deformation of faces and facial features. There are many techniques of feature extraction from an image. Here is a list of the methods:



Method	Notes
Principal Component Analysis (PCA)	Eigenvector-based, linear map
Kernel PCA	Eigenvector-based , non-linear map, uses kernel methods
Weighted PCA	PCA using weighted coefficients
Linear Discriminant Analysis (LDA)	Eigenvector-based, supervised linear map
Kernel LDA	LDA-based, uses kernel methods
Semi-supervised Discriminant Analysis (SDA)	Semi-supervised adaptation of LDA
Independent Component Analysis (ICA)	Linear map, separates non-Gaussian distributed features
Neural Network based methods	Diverse neural networks using PCA, etc.
Multidimensional Scaling (MDS)	Nonlinear map, sample size limited, noise sensitive.
Self-organizing map (SOM)	Nonlinear, based on a grid of neurons in the feature space
Active Shape Models (ASM)	Statistical method, searches boundaries
Active Appearance Models (AAM)	Evolution of ASM, uses shape and texture
Gavor wavelet transforms	Biologically motivated, linear filter
Discrete Cosine Transform (DCT)	Linear function, Fourier-related transform, usually used 2D-DCT
MMSD, SMSD	Methods using maximum scatter difference criterion.

Table 2 : Feature extraction algorithms.

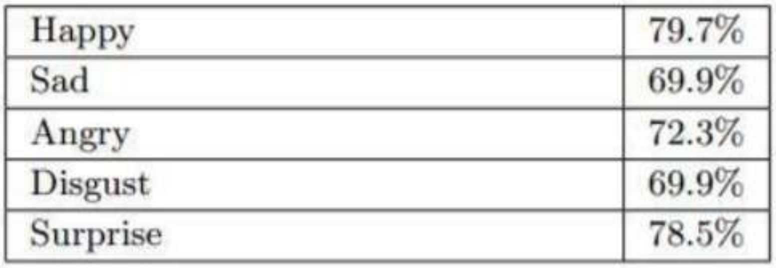
Among all these Algorithms I have tried to Work with Neural Network based methods, Gavor Wavelet Transforms, and Viola Jones Algorithms to find out the one that provide the most accurate face feature extraction and implement the System which will detect the emotion by reading those features. Here are some of the Performance evaluations of these Methods of Feature Extraction and Emotion Recognition using these.



PERFORMANCE ANALYSIS

No.	Type of Gesture	No. of Input Images	Recognized	Result (%)
1	Happy	13	12	92.3
2	Disgust	11	10	90.9
3	Anger	10	9	90
4	Neutral	7	7	100

Fig 8: performance analysis for Weighted PCA approach



Happy	79.7%
Sad	69.9%
Angry	72.3%
Disgust	69.9%
Surprise	78.5%

Fig 9: Performance analysis of gavor Wavelet Method

Despite there are these approaches which provides quite accurate output for the system, But I have used Viola Jones Algorithm to Extract Feature from the Image. Viola Jones object detection Framework is one of the modern algorithm which is based on Haar Cashcades which is an appearance based model. The Viola–Jones object detection framework is the first object detection framework to provide competitive object detection rates in real-time proposed in 2001 by Paul Viola and Michael Jones. Although it can be trained to detect a variety of object classes, it was motivated primarily by the problem of face detection. This approach to detecting objects in images combines four key concepts:

- Simple rectangular features, called Haar features.
- An Integral Image for rapid feature detection .
- The AdaBoost machine-learning method .
- A cascaded classifier to combine many features efficiently.

The features that Viola and Jones used are based on Haar wavelets. Haar wavelets are single wavelength square waves (one high interval and one low interval). In two dimensions, a square wave is a pair of adjacent rectangles - one light and one dark.

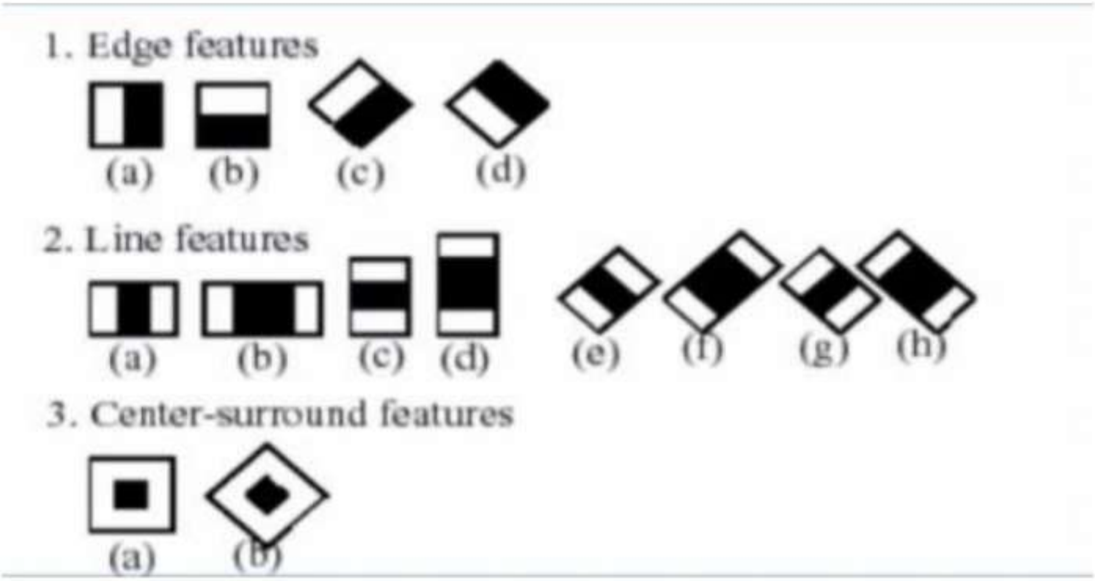


Fig 10: Haar-Like Feature

These are few types of Haar Classifier that were developed for detecting different type of features and for different environment. The actual rectangle combinations used for visual object detection are not true Haar wavlets. Instead, they contain rectangle combinations better suited to visual recognition tasks. Because of that difference, these features are called Haar features, or Haarlike features, rather than Haar wavelets. Figure above shows the features that OpenCV uses.

Viola and Jones combined a series of AdaBoost classifiers as a filter chain, shown in Figure below, that's especially efficient for classifying image regions. Each filter is a separate AdaBoost classifier with a fairly small number of weak classifiers.

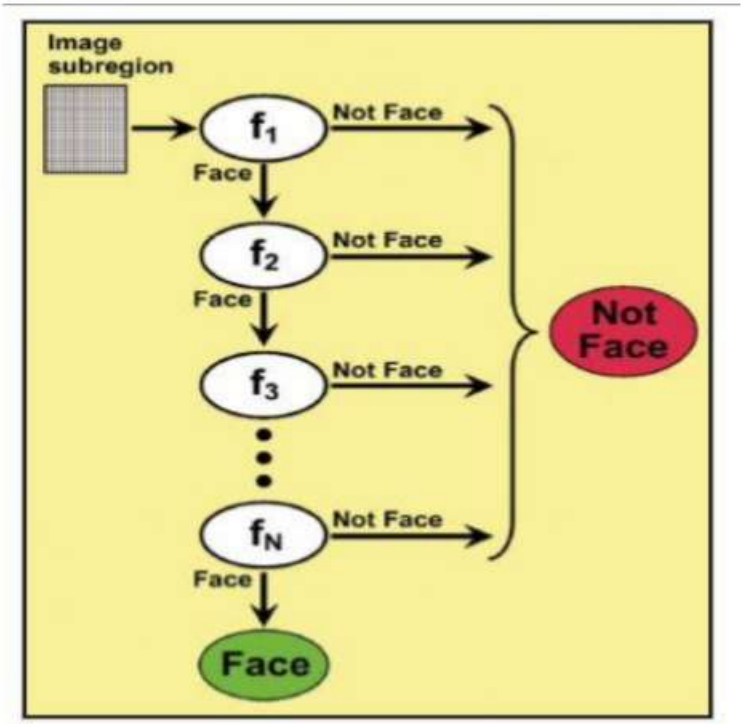
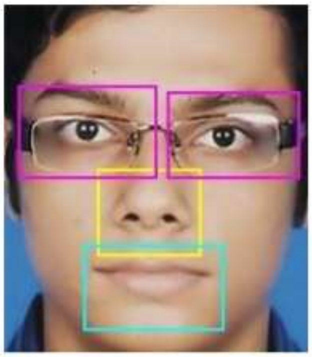


Fig 11: The classifer cascade is a chain of filters .Image subregions that make it through the entire cascade are classified as face all others not face

Here are some of the images where face features were detected by the System.



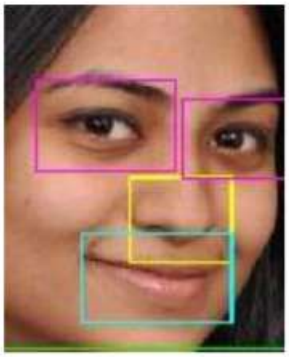


Fig 12: Detection of face Features Such as Eyes Nose and Lips.

4. Python pipeline

The dataset of 327 files was stored in a directory and each file was processed to create the feature set. As soon as the file was picked up, the name of the file was parsed to extract the emotion label. The emotion label was appended to a list of labels which will form our multi-class target variable. The image was processed for face detection and feature prediction. The features derived from each file were appended to a list which was later converted to a numpy array of dimension 327*68*2. We also had the target classes in the form of a numpy array. Same process was followed for RaFD database.

5. Machine learning

Once we had created the feature set and the target variable, we used Support Vector Machines to predict the emotions. Sklearn machine library was used to implement the Support Vector Machines (SVM) and Logistic Regression algorithms. The multiclass strategy used was ―One-Vs-Rest‖ for all the algorithms. Logistic regression algorithm was fine tuned for penalty ―l1‖ and ―l2‖. We also fine-tuned the linear kernel to rbf and poly to see the variation in results. Cross-validation technique was used along with SVM to remove any biases in the databases. Initially the dataset was divided as 70% for training and 30% for testing. We tried many other splits such as 80:20 and 70:30. 70:30 split seemed more appealing as our assumption was all classes will be equally represented in the test set. For cross-validation score we initially tested with 4 splits. To improve the results we chose the value 5 and 10, which are standard values for 29 cross-validation. Random Forest Classifier and Decision Trees were also run on our dataset, but resulted into low accuracy as compared to other algorithms in our experiment; hence we decided to continue with SVM and Logistic Regression.

Conclusion

The work done on emotion recognition and for achieving that all superior and novel approaches and methods. We have proposed a glimpse of a probable solution and method towards recognition the emotion. Work so far substantiate that emotion recognition using users EEG signal and audiovisual signal has the highest recognition rate and has highest performance. a research to classify facial emotions over static facial images using deep learning techniques was developed. This is a complex problem that has already been approached several times with different techniques.

Nowadays, facial emotion detection software includes the use of feature engineering. A solution totally based on feature learning does not seem close yet because of a major limitation: the lack of an extensive dataset of emotions. For instance, ImageNet contest uses a dataset containing 14 197 122 images. By having a larger dataset, networks with a larger capability to learn features could be implemented. Thus, emotion classification could be achieved by means of deep learning techniques.

Future scope

Nowadays, optimizers work on a stochastic basis because of large datasets (millions of examples). However, this was true for our project. From the real time emotion detection we can perform security safety . the code we have prepared which can be generated the security cameras where it can store the no. of humans persons are ined in a day and what type of emotion it can be stored . This behaviour can be displayed on the loss curve having a smoother shape or by avoiding an early convergence. We can perform the code in many ways like taking attendance in the online classes where it can store the all students database and directly it can be graded the attendance of the student.

Given a limited dataset, trying on the whole dataset could have lead to a better feature learning. Also, the use of some optimizers reported on this research would have had a different behaviour. Second, due to time constraints, it was not possible to evaluate each emotion. On this way, it would have been possible to detect which emotions are easier to classify, as well as, which ones are more difficult. Moreover, pre-training on each emotion could lead to a better feature learning. After that, the network could have received this learning (transfer learning). This could have improved on reducing the training time; as well as, minimizing to a higher degree the cost function.

Further improvement on the network's accuracy and generalization can be achieved through the following practices. The first one is to use the whole dataset during the optimization. Using batch optimization is more suitable for larger datasets. Another technique is to evaluate emotions one by one. This can lead to detect which emotions are more difficult to classify. Finally, using a larger dataset for training seems beneficial.

References

1. "Face Detection: Facial acknowledgment and discovering Homepage".

2. Lewis, Michael B; Ellis, Hadyn D (2003), "How we recognize a face: An overview of mental proof", International Journal of Imaging Systems and Technology, 13: 3–7, doi:10.1002/ima.10040, S2CID 14976176

3. Sheu, Jia-Shing; Hsieh, Tsu-Shien; Shou, Ho-Nien (1 December 2014). "Programmed Generation of Facial Expression Using Triangular Geometric Deformation". Diary of Applied Research and Technology. 12 (6): 1115–1130. doi:10.1016/S1665-6423(14)71671-2. ISSN 2448-6736.

4. Jun Zhang; Yong Yan; Lades, M. (1997). "Movement acknowledgment: Eigenface, flexible coordinating, and neural nets". Procedures of the IEEE. 85 (9): 1423–1435. doi:10.1109/5.628712.

5. Maity, Abhishek; Dasgupta, Sayan; Paul, Debjit (2015). "A Novel Approach to Face Detection utilizing Image Parsing and Morphological Analysis". Worldwide Journal of Computer Trends and Technology. 23 (4): 156–161.

6. "DCRP Review: Canon PowerShot S5 IS". Dcresource.com. Recovered 2011-02-15.

7. Tesco face discovery flashes unnecessary reconnaissance alarm, Facebook falls flat with teenagers, questions over Google+ | Technology | theguardian.com

8. IBM needs to manage the protection issue of facial acknowledgment | Technology | amarvelfox.com

9. http://en.wikipedia.org/wiki/RGB_color_model#Color_management
10. http://en.wikipedia.org/wiki/HSL_and_HSV#Use_in_image_analysis

11. www.stanford.edu/class/ee368/Project_03/.../ee368group16.ppt
12. http://en.wikipedia.org/wiki/Facial_Action_Coding_System
13. http://face-and-emotion.com/dataface/facs/manual/TitlePage.html
14. B. Fasel and J. Luettin. Automatic facial expression
15. analysis: A survey. Pattern Recognition, 2003.
16. www.ehu.es/ccwintco/uploads/e/eb/PFC-IonMarques.pdf
17. isl.ira.uka.de/downloads/icarcv2000_jie1.pdf
18. http://face-and-emotion.com/dataface/facs/manual/TOC.html
19. www.ijcte.org/papers/213-H282.pdf