**FACIAL ATTENDANCE SYSTEM**

**CONTENT**

ABSTRACTION………………………………………………..

1. LITERATURE REVIEW …………………………………….

2. INTRODUCTION ………………………………………………

2.1 PRIJECT AIMS ………………………………….

3. SCOPE OF THE PROJECT

4. FACIAL EMOTIONS APPLICATIONS …………………………………………..

5. PROJECT OVERVIEW …………………………..

6. MODEL IMPLEMENTATION ……………………………………

1. FACIAL EMOTION RECOGNITION C ………………………

2. IMAGE FEAATURES ……………………………

3. FACES …………………….

4. LANDMARKS ………………………

5. FEATURE DESCRIPTORS ………………………………

7. RELATED WORK ………………………….

1. FEATURE EXTRACTION TECHNIQUES ……………….

2. DISPLACEMENT RATIOS …………………………..

8. TOOLS AND LIBRARIES USED …………………………..

9. IMPLEMENTATION …………………………………

10. INSTALATION AND CODE IMPLEMENTATION………………………..

1 INSTALSTION PYTHON ………………..

2 INSTALL REQUIRD LIBRARIES …………………………………..

3 GATHER AND PREPARE INPUT DATA ………………………..

4 MODEL IMPLEMENTATON ……………………………….

11. CONCLUSION

**Abstract**

Facial emotion recognition (FER) is an emerging and significant research area in the pattern recognition domain. In daily life, the role of non-verbal communication is significant, and in overall communication, its involvement is around 55% to 93%. Facial emotion analysis is efficiently used in surveillance videos, expression analysis, gesture recognition, smart homes, computer games, depression treatment, patient monitoring, anxiety, detecting lies, psychoanalysis, paralinguistic communication, detecting operator fatigue and robotics. In this paper, we present a detailed review on FER. The literature is collected from different reputable research published during the current decade. This review is based on conventional machine learning (ML) and various deep learning (DL) approaches. Further, different FER datasets for evaluation metrics that are publicly available are discussed and compared with benchmark results. This paper provides a holistic review of FER using traditional ML and DL methods to highlight the future gap in this domain for new researchers. Finally, this review work is a guidebook and very helpful for young researchers in the FER area, providing a general understating and basic knowledge of the current state-of-the-art methods, and to experienced researchers looking for productive directions for future work.

**Keywords:** facial expressions; facial emotion recognition (FER); technological development; healthcare; security; deep learning and traditional classification methods

**1.LITERATURE REVIEW**

Some scholars have made efforts into the recognition of facial expressions. For instance, Fan and Tjahjadi proposed a facial expression recognition framework by combining CNN and handcrafted features. They found that the neural network could achieve brilliant recognition effects by extracting texture information from facial patches, and the introduction of CNN had a promoting effect on facial expression recognition [8]. Reddy et al. put forward an organic combination of deep learning features and the manual facial expression recognition method. They verified the applicability of this method in the wild scenes in experiments, which revealed the effectiveness of the recognition method under the combination of deep learning and manual production [9]. Liang et al. emphasized that the traditional handcrafted facial representations could only reveal shallow features. To break out this limitation, they proposed a new facial expression recognition method based on patches of interest, the Patch Attention Layer of embedding handcrafted features, to learn the local shallow features of each patch on face images. They finally proved the effectiveness of the method in their study [10]. Jain et al. (2018) built a hybrid convolutional-recursive neural network for facial expression recognition, which extracted information on face images through the combination of convolutional layer and Recurrent Neural Network (RNN) in view of the time correlations in images [11]. Avots et al. identified human emotions through analyzing audiovisual information. Besides, they classified the emotions on face images using different datasets as test sets, Viola-Jones face recognition algorithm, and CNN (AlexNet) [12]. Li et al. constructed a two-dimensional principal component analysis network via deep learning based on L1 norm for face recognition, tested it on some facial image database, and found that the network was robust [13]. Bernhard et al. stated that emotions had a significant influence on decision-making of humans, and hence, they applied deep learning to improve emotion recognition results. They found that the performance of RNNs and transfer learning was better than traditional machine learning, which had great inspirations on emotion computing application [14]. Kumar et al. discussed the modeling of abnormal facial expressions based on computer vision tasks and emotional deviations. They found that deep CNN could play an essential role in the training and classification of facial expressions, which provided a new visual modeling method for visual surveillance systems [15]. Mishra et al. employed CNN to recognize different emotions and intensity levels of human faces, which provided the basis and support for future research on computer emotion recognition [16].

In summary, the traditional manual methods are no longer suitable for the current research on facial expression recognition, but deep learning can significantly improve the recognition effectiveness of facial expression, especially the hybrid model. Although some achievements have been made in the previous studies of facial expression recognition, there are few studies combining it with psychological analysis.

**2.INTRODUCTON**

Facial emotions and their analysis play a vital role in non-verbal communication. It makes oral communication more efficient and conducive to understanding the concepts [1,2].

It is also conducive to detecting human attention, such as behavior, mental state, personality, crime tendency, lies, etc. Regardless of gender, nationality, culture and race, most people can recognize facial emotions easily. However, a challenging task is the automation of facial emotion detection and classification. The research community uses a few basic feelings, such as fear, aggression, upset and pleasure. However, differentiating between many feelings is very challenging for machines [3,4]. In addition, machines have to be trained well enough to understand the surrounding environment—specifically, an individual’s intentions. When machines are mentioned, this term includes robots and computers. A difference is that robots involve communication abilities to a more innovative extent since their design consists of some degree of autonomy [5,6]. The main problem is classifying people’s emotions is variations in gender, age, race, ethnicity and image quality or videos. It is necessary to provide a system capable of recognizing facial emotions with similar knowledge as possessed by humans. Recently, FER has become an emerging field of research, particularly for the last few decades. Computer vision techniques, AI, image processing and ML are widely used to expand effective automated facial recognition systems for security and healthcare applications [7,8,9,10].

Face detection is the first step of locating or detecting face(s) in a video or single image in the FER process. The images do not consist of faces only, but instead present with complex backgrounds. Indeed, human beings can easily predict facial emotions and other facial features of an image, but these are difficult tasks for machines without excellent training [11,12]. The primary purpose of face detection is to separate face images from the background (non-faces). Some face detection domains are gesture recognition, video surveillance systems, automated cameras, gender recognition, facial feature recognition, face recognition, tagging and teleconferencing [13,14]. These systems need first to detect faces as inputs. There is a color sensor for image acquisition that captures color images everywhere. Hence, current face recognition techniques depend heavily on grayscale, and there are only a few techniques that are able operate with color images. To achieve better performance, these systems either implement window-based or pixel-based techniques, which are the key categories of strategies for facial recognition. The pixel-based method lags in separating the face from the hands of another skin region of the person [15,16].

In contrast, the window-based method loses the capacity to view faces from various perspectives. Model matching approaches are the most frequently used techniques for FER, including face detection. In contrast, the window-based approach cannot view faces from different angles. Currently, several state-of-the-art classifiers, such as ANN, CNN, SVM, KNN, and random forest (RF), are employed for different features’ extraction and in the recognition of tumors in healthcare, in biometrics, in handwriting studies and in detecting faces for security measures [17,18,19].

**2.1PROJECT AIMS:**

Develop an accurate and reliable facial emotion recognition system.

Enhance the understanding of human emotions through facial expressions.

Improve human-computer interaction by enabling machines to recognize and respond to user emotions.

Facilitate mental health assessment by automating the analysis of facial expressions.

Enable personalized and adaptive systems that can tailor experiences based on detected emotions.

Contribute to the advancement of affective computing and artificial intelligence research.

**3. PROJECT SCOPE**

The scope of a facial emotion recognition project encompasses the specific boundaries and areas that the project aims to cover. It defines the extent of the project and outlines what will be included and excluded. The scope of a facial emotions recognition project may include:

**Data Collection:** The project may involve collecting or acquiring a diverse dataset of facial expressions labeled with corresponding emotions. The dataset should cover a wide range of individuals, demographics, and cultural backgrounds to ensure robustness and generalizability.

**Preprocessing and Data Cleaning:** Preprocessing steps may include image or video normalization, resizing, and noise reduction. Data cleaning ensures the removal of any biases, errors, or inconsistencies in the dataset that could impact the accuracy and reliability of the system.

**Face Detection:** The project may include implementing and applying face detection algorithms to accurately locate and extract facial regions from images or video frames. This step is essential for subsequent feature extraction and emotion classification.

**Feature Extraction:** Relevant facial features such as the position, shape, and movement of key facial landmarks may be extracted to represent the emotional state. This step involves using computer vision techniques to capture meaningful information from facial expressions.

**Emotion Classification:** Machine learning or deep learning algorithms may be employed to classify emotions based on the extracted facial features. The project may explore different classification approaches, such as support vector machines, decision trees, convolutional neural networks (CNNs), or recurrent neural networks (RNNs).

**Real-Time Analysis:** The project may include developing mechanisms or algorithms to enable real-time analysis of facial expressions. This allows for immediate feedback or response in applications that require real-time interaction, such as human-computer interfaces or virtual reality systems.

**Performance Evaluation:** The project should include evaluating the performance of the facial emotion recognition system using appropriate metrics and benchmark datasets. This step helps assess the accuracy, precision, recall, and overall performance of the developed system.

**Robustness and Adaptability:** The project may focus on addressing challenges related to variations in lighting conditions, pose, occlusions, and cultural influences. Developing a system that can handle these variations ensures robustness and adaptability across different scenarios and individuals.

**Documentation and Reporting:** The project should produce a comprehensive report or research paper documenting the project's methodologies, findings, and results. Clear documentation enables reproducibility and sharing of knowledge with the research community.

**Potential Applications:** The project may explore and discuss potential applications of facial emotion recognition, such as human-computer interaction, affective computing, mental health assessment, or entertainment.

It's important to note that the scope may vary depending on the resources, time constraints, and objectives of the specific project. Defining a clear scope helps in managing project expectations, setting realistic goals, and ensuring a focused and successful facial emotion recognition project.

**4. FACIAL EMOTIONS RECOGNITION APPLICATIONS**

In the past two years, emotion AI vendors have moved into completely new areas and industries, helping organizations to create a better customer experience and unlock real cost savings. These uses include:

1. **Video gaming.** Using computer vision, the game console/video game detects emotions via facial expressions during the game and adapts to it.
2. **Medical diagnosis**. Software can help doctors with the diagnosis of diseases such as depression and dementia by using voice analysis.
3. **Education.** Learning software prototypes have been developed to adapt to kids’ emotions. When the child shows frustration because a task is too difficult or too simple, the program adapts the task so it becomes less or more challenging. Another learning system helps autistic children recognize other people's emotions.
4. **Employee safety.** Based on Gartner client inquiries, demand for employee safety solutions are on the rise. Emotion AI can help to analyze the stress and anxiety levels of employees who have very demanding jobs such as first responders.
5. **Patient care.** A ‘nurse bot’ not only reminds older patients on long-term medical programs to take their medication, but also converses with them every day to monitor the their overall wellbeing.
6. **Car safety.** Automotive vendors can use computer vision technology to monitor the driver's emotional state. An extreme emotional state or drowsiness could trigger an alert for the driver.
7. **Autonomous car.** In the future, the interior of [**autonomous cars**](https://www.gartner.com/smarterwithgartner/4-areas-driving-autonomous-vehicle-adoption/) will have many sensors, including cameras and microphones, to monitor what is happening and to understand how users view the driving experience.
8. **Fraud detection.** Insurance companies use voice analysis to detect whether a customer is telling the truth when submitting a claim. According to independent surveys, up to 30% of users have admitted to lying to their car insurance company in order to gain coverage.
9. **Recruiting.** Software is used during job interviews to understand the credibility of a candidate.
10. **Call center intelligent routing.** An angry customer can be detected from the beginning and can be routed to a well-trained agent who can also monitor in real-time how the conversation is going and adjust.
11. **Connected home.** A [**VPA-enabled speaker**](https://www.gartner.com/en/newsroom/press-releases/2017-08-24-gartner-says-worldwide-spending-on-vpa-enabled-wireless-speakers-will-top-3-billion-by-2021) can recognize the mood of the person interacting with it and respond accordingly.
12. **Public service.** Partnerships between emotion AI technology vendors and surveillance camera providers have emerged. Cameras in public places in the United Arabic Emirates can detect people's facial expressions and, hence, understand the general mood of the population. This project was initiated by the country's Ministry of Happiness.
13. **Retail.** Retailers have started looking into installing computer vision emotion AI technology in stores to capture demographic information and visitors' mood and reactions

**5. PROJECT OVERVIEW**

**5.1.Emotion Detection:** The primary objective of the project is to develop the robust algorithms and models that can accurately detect and organize the facial expressions associated with different emotions. This involves training machine learning and computer vision models on large datasets of labelled facial expression to enable them to effectively classify emotions such as happiness, sadness, anger, surprise, disgust, wink, Gray pout, fear, neutral.

**5.2. Feature Extraction:** Another objective is to identify and extract relevant facial features and patterns that are indicative of specific emotions. This involves analysing facial muscle movements, changes in facial landmarks, and other visual cues to capture the distinctive characteristics of different emotional expressions.

**5.3. Accuracy and Reliability:** The project aims to achieve high accuracy and reliability in emotion recognition. This involves refining the algorithms and models through iterative training and testing, continuously improving their performance to minimize false positives and false negatives in emotion classification.

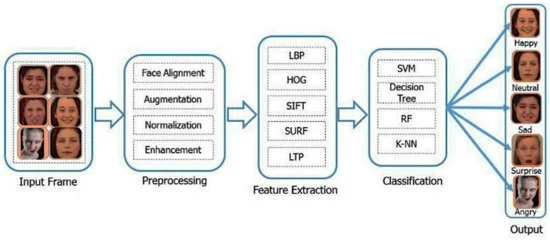
**5.4. Real-Time Processing:** The project seeks to develop algorithms that can process facial emotions in real-time. This is important for applications such as human-computer interaction, where timely responses based on detected emotions are required.

**5.5 Application Integration:** The project goals include integrating the developed facial emotion recognition technology into various applications and systems. This involves creating APIs, software libraries, or standalone applications that can be easily integrated into different domains such as virtual reality, gaming, market research, healthcare, and more.

**6.MODEL IMPLEMENTATION**

**6.1 Facial Emotion Recognition Using Traditional Machine Learning Approaches**

Facial emotions are beneficial for investigating human behavior [23,24] as exhibited in Figure 1. Psychologically, it is proven that the facial emotion recognition process measures the eyes, nose, mouth and their locations.



**Figure 1. Facial emotion recognition (FER) process.**

The earliest approach used for facial emotion intensity estimation was based on distance urged. This approach uses high-dimensional rate transformation and regional volumetric distinction maps to categorize and quantify facial expressions. In videos, most systems use Principal Component Analysis (PCA) to represent facial expression features [25]. PCA has been used to recognize the action unit to express and establish different facial expressions. Other facial expressions are structured and recognized by mistreatment PCA for providing a facial action unit [26].

Siddiqi et al. [27] detected and extracted the face portion via the active contour model. The researchers used Chan–Vese and Bhattacharyya’s energy functions to optimize the distance between face and context, and reduce the differences within the face. In addition, noise is reduced using wavelet decomposition, and the geometric appearance features of facial emotions and facial movement features using optical flow are extracted.

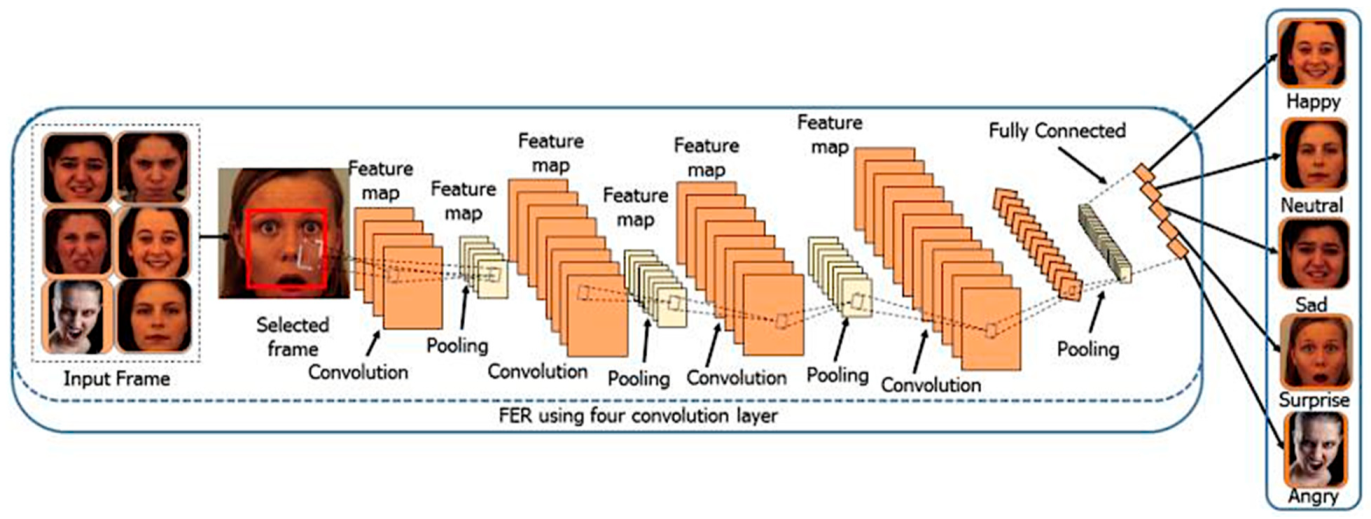
There is no need for high computational power and memory for conventional ML methods such as DL methods. Therefore, these methods need further consideration to implement embedded devices that perform classification in real time with low computational power and provide satisfactory results. Accordingly, Table 1 presents a brief summary

**Table 1.** Summary of the representative conventional FER approaches.

| **References** | **Datasets** | **Decision Methods** | **Features** | **Emotions Analyzed** |
| --- | --- | --- | --- | --- |
| Varma et al. [[**28**](https://www.mdpi.com/2078-2489/13/6/268#B28-information-13-00268)] | FECR | support vector machine (SVM) classifier and HMM | PCA and LDA | Six emotions |
| Reddy et al. [[**29**](https://www.mdpi.com/2078-2489/13/6/268#B29-information-13-00268)] | CK+ | support vector machine (SVM) classifier | Haar Wavelet Transform (HWT), Gabor wavelets and nonlinear principal component analysis (NLPCA) | Six emotions |
| Sajjad et al. [[**30**](https://www.mdpi.com/2078-2489/13/6/268#B30-information-13-00268)] | MMI JAFFE CK+ | support vector machine (SVM) classifier | ORB SIFT SURF | Seven emotions |
| Nazir et al. [[**31**](https://www.mdpi.com/2078-2489/13/6/268#B31-information-13-00268)] | CK+ MMI | KNN, SMO and MLP for classification | HOG, DCT | Seven emotions |
| Zeng et al. [[**32**](https://www.mdpi.com/2078-2489/13/6/268#B32-information-13-00268)] | CK+ | DSAE | LBP, SIFT, Gabor Function and HOG | Seven emotions |
| Uddin et al. [[**33**](https://www.mdpi.com/2078-2489/13/6/268#B33-information-13-00268)] | Real-time dataset from visual depth camera | Deep Belief Network (DBN) | MLDP-GDA features | Six emotions |
| Al-Agha et al. [[**34**](https://www.mdpi.com/2078-2489/13/6/268#B34-information-13-00268)] | BOSPHORUS | Euclidean distance | Geometric descriptor | Four emotions |
| Ghimire et al. [[**35**](https://www.mdpi.com/2078-2489/13/6/268#B35-information-13-00268)] | CK+ MMI MUG | EBGM, KLT and AdaBoost-ELM | Salient geometric features | Seven emotions |
| Wang and Yang [[**36**](https://www.mdpi.com/2078-2489/13/6/268#B36-information-13-00268)] | BVTKFER BCurtinFaces | random forest classifier | LBP | Six emotions |
| Wu et al. [[**37**](https://www.mdpi.com/2078-2489/13/6/268#B37-information-13-00268)] | BBC, SPOS MMI UvANEMO | support vector machine (SVM) | RSTD, four conventional features—raw pixels, Gabor, HOG and LBP | Two emotions Smile (genuine and fake) |
| Acevedo et al. [[**38**](https://www.mdpi.com/2078-2489/13/6/268#B38-information-13-00268)] | CK+ | Conditional Random Field (CRF) and KNN | Geometric descriptor | Seven emotions |
| Kim [[**39**](https://www.mdpi.com/2078-2489/13/6/268#B39-information-13-00268)] | CK JAFFE | embedded hidden Markov model (EHMM) | ASM and 2D DCT | Seven emotions |
| Cornejo et al. [[**25**](https://www.mdpi.com/2078-2489/13/6/268#B25-information-13-00268)] | CK+ JAFFE MUG | PCA, LDA, K-NN and SVM | Gabor wavelets and geometric features | Seven emotions |
| Siddiqi et al. [[**27**](https://www.mdpi.com/2078-2489/13/6/268#B27-information-13-00268)] | CK JAFFE USTCNVIE Yale FEI | hidden Markov model (HMM) | Chan–Vese energy function, Bhattacharyya distance function wavelet decomposition and SWLDA | Six emotions |
| Chang et al. [[**40**](https://www.mdpi.com/2078-2489/13/6/268#B40-information-13-00268)] | CK+ | Support Vector Regression (SVR) | feature descriptors, AAMs, Gabor wavelets | Seven emotions |

**6.1.1 Facial Emotion Recognition Using Deep-Learning-Based Approaches**

Deep learning (DL) algorithms have revolutionized the computer vision field in the current decade with RNN and CNN [41,42,43]. These DL-based methods are used for feature extraction, recognition and classification tasks. The key advantage of a DL approach (CNN) is to overcome the dependency on physics-based models and reduce the effort required in preprocessing and feature extraction phases [44,45]. In addition, DL methods enable end-to-end learning from input images directly. For these purposes, in several regions, including FER, scene awareness, face recognition and entity recognition, DL-based methods have obtained encouraging results from the state-of-the-art [46,47]. There are generally three layers in a DL-CNN, (1) convolution layer, (2) subsampling layer and (3) FC layer, as exhibited in Figure 2. The CNN takes the image or feature maps as the input, and slides these inputs together with a series of filter banks to produce feature maps that reflect the facial image’s spatial structure. Inside a feature map, the weights of convolutional filters are connected, and the feature map layer inputs are locally connected [48,49,50]. By implementing one of the most popular pooling approaches, i.e., max pooling, min pooling or average pooling, the second type of layer, called subsampling, is responsible for reducing the given feature maps [51,52]. A CNN architecture’s last FC layer calculates the class probability of an entire input image. Most DL-based techniques can be freely adapted with a CNN to detect emotions.



**Figure 2. Training process of CNN model for facial emotion recognition.**

Li et al. [53] proposed a 3D CNN architecture to recognize several emotions from videos. They extracted deep features and used three benchmark datasets for the experimental evaluation, namely CASME II, CASME and SMIC. Li et al. [54] performed additional face cropping and rotation techniques for feature extraction using a convolutional neural network (CNN). Tests were carried out on the CK+ and JAFFE databases to test the proposed procedure.

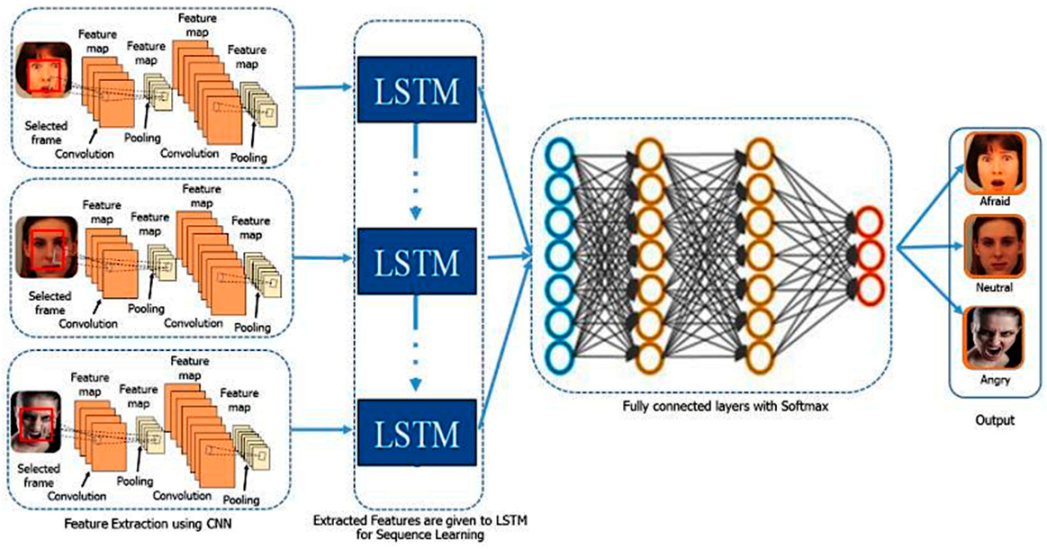
A convolution neural network (CNN) with a focus function (ACNN) was suggested by Li et al. [55] that could interpret the occlusion regions of the face and concentrate on more discriminatory, non-occluded regions. A CNN is an end-to-end learning system. First, the different depictions of facial regions of interest (ROIs) are merged. Then, each representation is weighted by a proposed gate unit that calculates an adaptive weight according to the area’s significance. Two versions of ACNN have been developed in separate ROIs: patch-based CNN (pACNN) and global-local-based ACNN (gACNN). Lopes et al. [56] classified human faces into several emotion groups. They used three different architectures for classification: (1) a CNN with 5 C layers, (2) a baseline with one C layer and (3) a deeper CNN with several C layers. Breuer and Kimmel [57] trained a model using various datasets of FER to classify seven basic emotions. Chu et al. [58] presented multi-level mechanisms for detecting facial emotions by combining temporal and spatial features. They used a CNN architecture for spatial feature extraction and LSTMs to model the temporal dependencies. Finally, they fused the output of both LSTMs and CNNs to provide a per-frame prediction of twelve facial emotions. Hasani and Mahoor [59] presented the 3D Inception-ResNet model and LSTM unit, which were fused to extract temporal and spatial features from the input frames of video sequences. (Zhang et al., [60] and Jain et al. [61] suggested a multi-angle optimal pattern-dependent DL (MAOP-DL) system to address the problem of abrupt shifts in lighting and achieved proper alignment of the feature set by utilizing optimal arrangements centered on multi-angles. Their approach first subtracts the backdrop, isolates the subject from the face images and later removes the facial points’ texture patterns and the related main features. The related features are extracted and fed into an LSTM-CNN for facial expression prediction.

Al-Shabi et al. [62] qualified and collected a minimal sample of data for a model combination of CNN and SIFT features for facial expression research. A hybrid methodology was used to construct an efficient classification model, integrating CNN and SIFT functionality. Jung et al. [63] proposed a system in which two CNN models with different characteristics were used. Firstly, presence features were extracted from images, and secondly, temporal geometry features were extracted from the temporal facial landmark points. These models were fused into a novel integration scheme to increase FER efficiency. Yu and Zhang [64] used a hybrid CNN to execute FER and, in 2015, obtained state-of-the-art outcomes in FER. They used an assembly of CNNs with five convolution layers for each facet word. Their method imposed transformation on the input image in the training process, while their model produced predictions for each subject’s multiple emotions in the testing phase. They used stochastic pooling to deliver optimal efficiency, rather than utilizing peak pooling (Table 2).

**Table 2. Deep learning-based approaches for FER.**

Table

The hybrid CNN-RNN and CNN-LSTM techniques have comparable architectures, as discussed in the previous section and exhibited in Figure 3. In short, CNN-RNN’s simple architecture combines an LSTM with a DL software visual feature extractor, such as the CNN model. The hybrid techniques are, thus, equipped to distinguish emotions from image sequences. Figure 3 indicates that each graphic attribute has been translated to the LSTM blocks and describes a variable or fixed-length vector. Finally, performance is given for the prediction in a recurrent sequence learning module and the SoftMax classifier is used in [58].



**3. Demonstrated CNN features used by LSTM for sequence learning; SoftMax is used.**

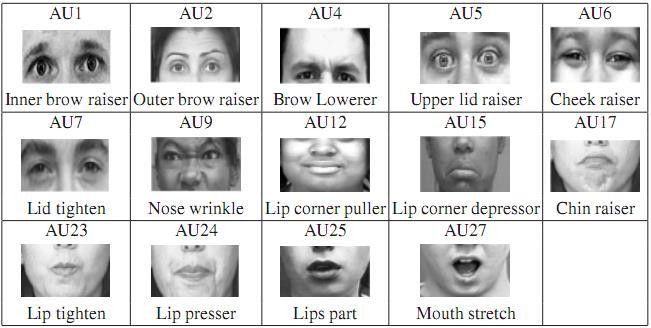
Generally, DL-based methods determine classifiers and features by DNN experts, unlike traditional ML methods. DL-based methods extract useful features directly from training data using DCNNs [67,68]. However, the massive training data are a challenge for facial expressions under different conditions to train DNNs. Furthermore, DL-based methods need high computational power and a large amount of memory to train and test the model compared to traditional ML methods. Thus, it is necessary to decrease the computational time during the inferencing of DL methods.

**6.2 Image Features**

We can derive different types of features from the image and normalize it in vector form. We can employ various types of techniques to identify the emotion like calculating the ellipses formed on the face or the angles between different parts like eyes, mouth etc. Following are some of the prominent features which can be used for training machine learning algorithms:

**6.3 Faces**

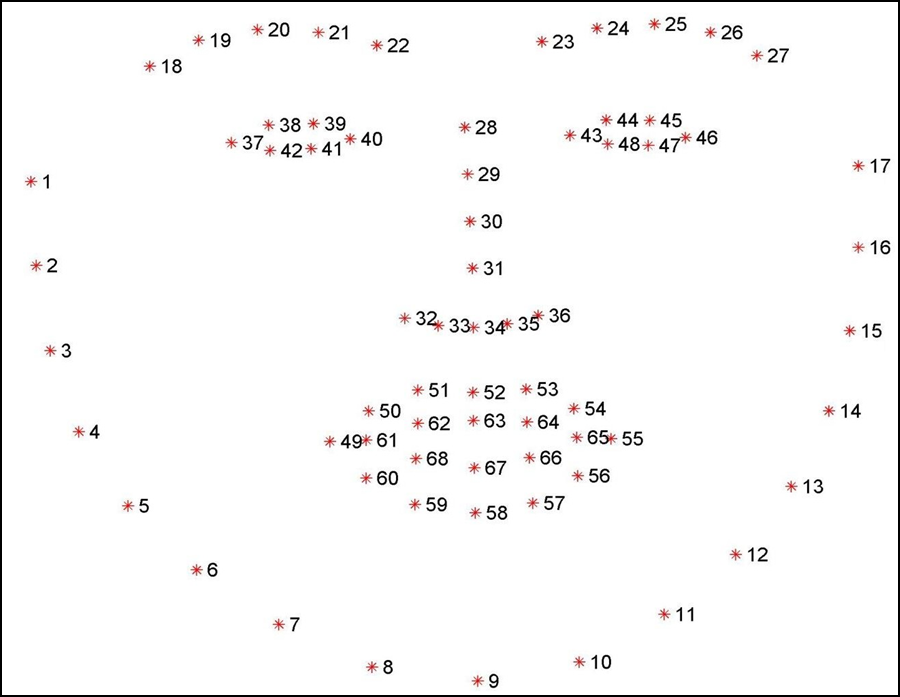
Facial Action Coding System is used to give a number to facial moment. Each such number is called as action unit. Combination of action units result in a facial expression. The micro changes in the muscles of the face can be defined by an action unit. For example, a smiling face can be defined in terms of action units as 6 + 12, which simply means movement of AU6 muscle and AU12 muscle results in a happy face. Here Action Unit 6 is cheek raiser and Action Unit 12 is lip corner puller. Facial action coding system based on action units is a good system to determine which facial muscles are involved in which expression. Real time face models can be generated based on them



**Figure 1:** **Action Units corresponding to different movements in face**

**6.4 Landmarks**

Landmarks on the face are very crucial and can be used for face detection and recognition. The same landmarks can also be used in the case of expressions. The Dlib library has a 68 facial landmark detector which gives the position of 68 landmarks on the face.



**Figure 2****: Landmarks on face**

Figure 2 shows all the 68 landmarks on face. Using dlib library we can extract the co- ordinates(x,y) of each of the facial points. These 68 points can be divided into specific areas like left eye, right eye, left eyebrow, right eyebrow, mouth, nose and jaw.

**6.5 Feature Descriptors**

Good features are those which help in identifying the object properly. Usually the images are identified on the basis of corners and edges. For finding corners and edges in images, we have many feature detector algorithms in the OpenCV library such as Harris corner detector.

These feature detectors take into account many more factors such as contours, hull and convex. The Key-points are corner points or edges detected by the feature detector algorithm. The feature descriptor describes the area surrounding the key-point. The description can be anything including raw pixel intensities or co-ordinates of the surrounding area. The key-point and descriptor together form a local feature. One example of a feature descriptor is a histogram of oriented gradients. ORB (based on BRIEF), SURF, SIFT etc. are some of the feature descriptor algorithms

**7. RELATED WORK**

3.1 Feature Extraction Techniques

This method uses cascaded regression trees and finds the important positions on the face using images. Pixel intensities are used to distinguish between different parts of the face, identifying 68 facial landmarks. Based on a current estimate of shape, parameter estimation is done by transforming the image in the normal co-ordinate system instead of global. Extracted features are used to re-estimate the shape parameter vectors and are recalculated until convergence



**Figure 3****: Image with19 feature points**

Here uses only 19 features as shown in Figure 3 from the 68 extracted features, focusing only around eyes, mouth and eyebrows

Ensemble of regression trees was very fast and robust giving 68 features in around 3 milliseconds. The parameters tuned for the algorithm are shown in Table 1

**Table 1: Parameters for tuning ensemble of regression trees algorithm**

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Cascade depth | 15 |
| Tree depth | 4 |
| Number of trees per  cascade level | 500 |
| Number of test splits | 20 |

### 7.2 Displacement ratios

Once the features are in place, the displacement ratios of these 19 feature points are calculated using pixel coordinates. Displacement ratios are nothing but the difference in pixel position in the image from initial expression to final expression. Twelve types of distances are calculated as shown in Table 2 [1].

Instead of using these distances directly, displacement ratios are used as these pixel distances may vary depending on the distance between the camera and the person.

The dataset used for this experiment was the iBug-300W dataset which has more than 7000 images along with CK + dataset having 593 sequences of facial expressions of 123 different subjects.

|  |  |
| --- | --- |
| Distance | Description of the distances |
| D1 and D2 | Distance between the upper and lower eyelid of the right and left eyes |
| D3 | Distance between the inner points of the left and right eyebrow |
| D4 and D5 | Distance between the nose point and the inner point of the left and right eyebrow |
| D6 and D8 | Distance between the nose point and the right and left mouth corner |
| D7 and D9 | Distance between the nose point and the midpoint of the upper and lower lip |
| D10 | Distance between the right and left mouth corner |
| D11 | Distance between the midpoint of the upper and lower lip |
| D12 | Mouth circumference |

**Face detection using viols-jones Face detector**

Images are pre-processed to reduce noise. Also, dataset must have images with equal level of exposure, illumination and brightness. Image enhancement is performed on such images using Histogram equalization techniques. The face detection module is the familiar Viola- Jones technique which uses Ada-boost algorithm. It combines a lot of weak classifiers to form a strong classifier by iterations, where a weak classifier reduces the weighted error rate per iteration

**Lbp technique for feature extraction**

Local Binary Pattern (LBP) is a very simple and robust technique for feature extraction and is independent of the illumination differences in the images. A 3x3 matrix is generated and each of the pixels in the matrix is assigned a binary value depending on the center pixel value. These 8- bit binary values form an 8 bit binary number excluding the center pixel value which is finally converted to decimal value

LBP code for a pixel at (xc, yc) is given by [6]

LBPP,R (xc, yc) = ∑(P=0,7) S(gp−gc) 2P, S(x) = {1, x >= 0 and 0, x < 0}

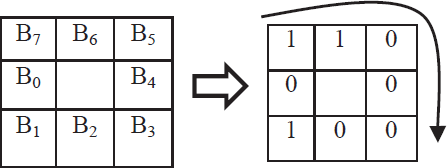
where,

gc = gray value of center pixel

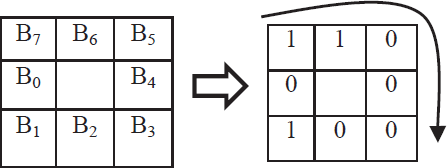
gp = gray value of neighboring pixel of gc

P = 8 (maximum 8 neighbors for center pixel in a 3x3 matrix) Hence a pixel can have 256 different values as 2P =28=256.

Figure 6 shows an input image block with center pixel value as 7. Each value of the surrounding 8 pixels is reassigned by comparing it to the intensity of the center pixel. Pixel values greater than or equal to the center pixel are assigned 1; otherwise 0. Figure 7 shows the binary values assigned to all the 8 pixels, which combined in clockwise direction, gives a binary 8 bit number. Converting the 8-bit number in the Figure 6 into decimal value (11000010) gives us the number 194 [6]



**Figure 6****: 3x3 initial input matrix [6]**



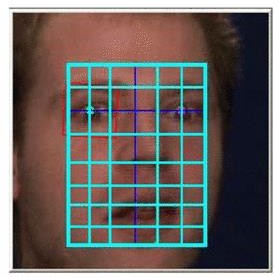
**Figure 7****: Pixel encoding using LBP [6]**

We can encode all such pixel blocks in an image and use the decimal values obtained as image feature vectors for classification. LBP is simple to implement as the binary pattern remains the same in spite of changes in the illumination or brightness. An increase in brightness or illumination condition will result in all the pixel intensities increasing by the same value, keeping the relative difference the same [6].

**Using dynamic grid – based HoG features**

Histogram of oriented gradient (HoG) is an object detection technique based on edge information. Each detection window can be defined by edge orientations. Image is cropped according to the area of interest on the face. This cropped face is the detection window which is divided into even smaller set of regions called cells. In each cell, for each pixel, a magnitude of edge gradient is computed for each orientation bin, thus forming the local histogram of oriented gradients [7].

As Figure 8 shows, from the face, required region is cropped and divided into a matrix of 8 rows x 6 columns. Pixel size may vary in the grid for each cell



**Figure 8****: Cropped image divided into cells of 8 rows x 6 columns [7]**

A local histogram is generated for each cell. Further, for each block of 2 x 2 cells, 4 histograms are concatenated. Finally we have 12 normalized histograms concatenated into one single global histogram which gives us 432 feature vectors [7].

### Geometrical facial features extraction

A set of 19 features are selected empirically by observing the landmark positions on the face and which are more related to human expressions. These 19 features are taken as a subset from an existing marker-less system for landmark identification and localization, which has actual 66 2D features [9] [10]. These 19 features or landmarks on the face are given in Table 3 and Figure 9. Landmark positions in the image space are used to define two set of features: eccentricity features and linear features [9].

### Eccentricity features

Eccentricity features are based on the concept of ellipses. The eccentricity of ellipses is the amount of deviation of the ellipse from being a circle. Eccentricity is between 0 and 1 for ellipses and 0 if the ellipse is a circle. For example, while smiling, the eccentricity will be greater than 0; but while expressing surprise it will be closer to 0 [9].

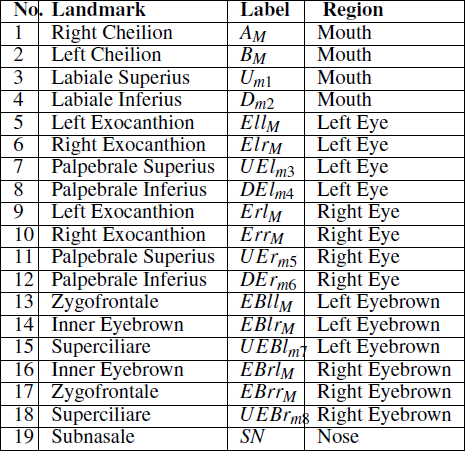
Refer to Figure 10. Here, AM and BM are the end-points of the major-axis which is the length of the mouth, whereas Um1 and Dm2 are the upper and lower end points of the minor axis respectively. The eccentricity is calculated for the upper and lower half separately. For the upper ellipse (AM, BM, Um1) eccentricity is given by,

e = √a2 – b2 / a

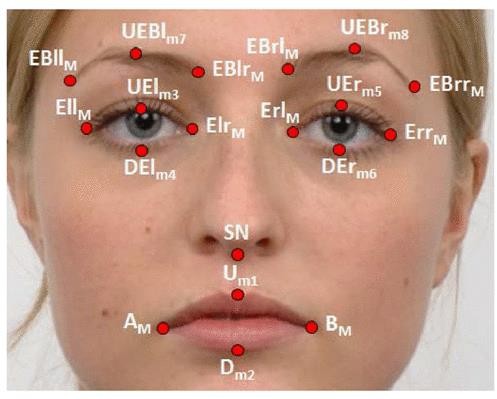
where,

a = BMx – AMx / 2

b = AMy – Um1y

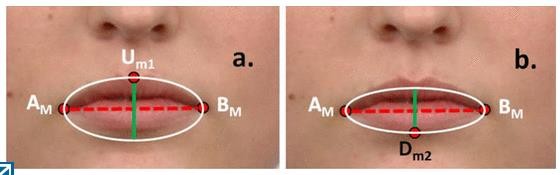


**Table 3****: 19 landmark points [9]**



**Figure 9****: Position of 19 landmark points on face [9**

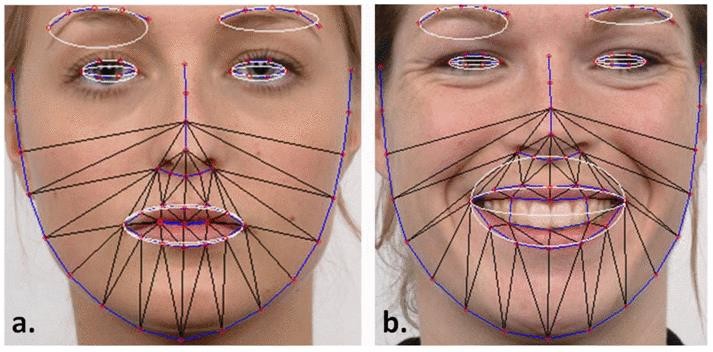
**Eccentricity extraction algorithm**



**Figure 10****: Ellipse for the upper part of mouth (a). Ellipse for lower part of mouth (b) [9]**

In the same way eccentricity is calculated for the other 7 ellipses: lower mouth, upper left eye, lower left eye, upper right eye, lower right eye, left eyebrow and right eyebrow. In Figure 11, part a shows eight ellipses, whereas part b shows the change in eccentricity as the person smiles.

**Figure 11:** **ellipses construction for different areas on face(a).changes in ellipse eccentricity(b) [9]**



**Linear Features**

Movements that occur during expressing emotions between facial landmarks can be quantitatively evaluated using distances. These are the normalized linear distances.

The following distances are calculated to determine the linear feature vectors [9]:

L1 – Movements between eyes and eyebrows

L2 – Movements between mouth and nose

L3 – Movements between upper and lower mouth points

**Machine Learning Algorithms**

Once the dataset is created using the required features, the next important step is to use a good classification algorithm. Support Vector Machines (SVM) are used almost in all cases of multi-class classification of human expressions [1][7][8]. They are combined with one or another feature extraction technique [8].

**Hidden Markov Models (HMM)**

Hidden Markov Models (HMM) is based on statistics, and are useful for finding hidden structure of data. They are also very popular for emotion detection through speech [12]. Input is a sequence of observed features and there are hidden states corresponding to consecutive events. HMM is expressed as follows [14]:

λ = (A,B,π)

where,

A = (aij) transition probability matrix between the hidden states

B = (bij) observation symbols probability from a state Π = initial probability of states.

Paper [13] describes the process of developing Code – HMM. It tries to improve on the existing HMM by incorporating some characteristics of multiple classifiers. Also, HMM are used in sequence with algorithms such as k-Nearest Neighbor [13]. Advantage of using both the methods is HMM can do the complex computations and k-NN just have to classify between the given samples. HMM decision is based on biggest output probability which might be mixed with

noise, whereas K-NN can add a second layer of classification thereby increasing accuracy.

HMM‘s also used in combination with SVM as Serial Multiple Classifier System to get best results for speech emotion recognition [12]. As SVM directly gives a classification instead of a score, HMM‘s can be used for training the samples and SVM for classification. Along with multiple classifiers, boosting can also be used as a means of developing a strong classification system where two or more weak classifiers are combined to form a strong classifier [15]. The paper [15] also talks about embedding HMM, i.e. developing a two-dimensional HMM, consisting of super states and embedded states. The data is modeled in two directions by super states and embedded states. For face images, top to bottom features can be super states and right –to-left features can be embedded states.

**Other Algorithms**

Random Forest Classifiers have also proven to have an upper hand over SVM in some of the cases [9]. Random forests are based on decision trees, but instead of just one classifier, use more forests or classifiers to decide the class of the target variable. Table 4 gives us the result of random forest classifier for detecting 6,7 and 8 emotions. Here, S1 to S5 represent the subset of emotion used for detection.

K-Nearest Neighbor, Linear Discriminant Analysis and Neural Networks (ANN) are some of the algorithms used for classification of prediction of emotion.



**8. TOOLS AND LIBRARIES USED**

**OpenCV**

OpenCV is the library we will be using for image transformation functions such as converting the image to grayscale. It is an open source library and can be used for many image functions and has a wide variety of algorithm implementations. C++ and Python are the languages supported by OpenCV. It is a complete package which can be used with other libraries to form a pipeline for any image extraction or detection framework. The range of functions it supports is enormous, and it also includes algorithms to extract feature descriptors

**Dlib**

Dlib is another powerful image-processing library which can be used in conjunction with Python, C++ and other tools. The main function this library provides is of detecting faces, extracting features, matching features etc. It has also support for other domains like machine learning, threading, GUI and networking.

**Python**

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 pre- processing. 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.

**Scikit-learn**

Scikit-learn is the machine learning library in python. It comprises of matplotlib, numpy and a wide array of machine learning algorithms. The API is very easy to use and understand. It has many functions to analyze and plot the data. A good feature set can be formed using many of its feature reduction, feature importance and feature selection functions. The algorithm it provides can be used for classification and regression problems and their sub-types.

**Jupyter NoteBook**

Jupyter Notebook is the IDE to combine python with all the libraries we will be using in our implementation. It is interactive, although some complex computations require time to complete. Plots and images are displayed instantly. It can be used as a one stop for all our requirements, and most of the libraries like Dlib, OpenCV, Scikit-learn can be integrated easily.

**9. IMPLEMENTATION**

A static approach using extracted features and emotion recognition using machine learning is used in this work. The focus is on extracting features using python and image processing libraries and using machine learning algorithms for prediction. Our implementation is divided into three parts. The first part is image pre-processing and face detection. For face detection, inbuilt methods available in dlib library are used. Once the face is detected, the region of interest and important facial features are extracted from it. There are various features which can be used for emotion detection. In this work, the focus is on facial points around the eyes, mouth, eyebrows etc.

We have a multi-class classification problem and not multi-label. There is a subtle difference as a set of features can belong to many labels but only one unique class. The extracted facial features along with SVM are used to detect the multi-class emotions. The papers we have studied focus on SVM as one of the widely used and accepted algorithms for emotion classification. Our database has a total of 7 classes to classify. We have compared our results with logistic regression and random forest to compare the results of different algorithms. The processing pipeline can be visualized as Figure 13.

**Image processing pipeline**

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.



**Figure 15:** **Original image from the database and detected face from the image**

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

descriptors along with SVM are used to identify the face from the image. Images are converted to grayscale and resized for uniformity.

**Facial feature extraction**

For facial feature extraction, we used the 68 landmark facial feature predictor from dlib. The face detector algorithm returns a window(x,y,width,height) which is the detected face. The detected face is passed to the feature predictor algorithm. Figure 16 shows the detected 68 landmarks for a particular face. The predictor function returns the 68 points at the eyes(left and right), mouth, eyebrows(eft and right), nose and jaw. We used numpy array to convert the 68 points to an array of 68 x and y co-ordinates representing their location. These are the facial features we have used to predict emotion.



**Figure 16****: Detected landmarks from the face**

The landmarks are easier to access in numpy array form. Also, from Figure 16 we know the indices of each feature, hence we can focus on a particular feature instead of the entire set.

The feature points are divided as 1-17 for jaw, 49-68 for mouth and so on. So, for instance, if we want to ignore the jaw, we can simply put the x and y co-ordinates for the jaw as 0, while converting the features into numpy array. We also calculated distances and polygon areas for some of the facial landmarks.

**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.

**Machine learning**

Once we had created the feature set and the target variable, we used CNN Machines to predict the emotions. Sklearn machine library was used to implement the (CNN) 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

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 convolutional neural network.

We applied convolutional neural network to our dataset and predicted the result. The results were interpreted using accuracy metric. The train and validation split were 75 and 25. The results are as follows:

|  |  |
| --- | --- |
| CNN | ACCURACY |
| Sequential | 66 |
|  |  |

**Distances :**

As we had the co-ordinates we could easily calculate the distance between any two facial points on the face using the distance formula:



√(𝑥2 − 𝑥1)2 + (𝑦2 − 𝑦1)2

In Figure 18 we show some of the horizontal and vertical distances calculated for the face. In all, we calculated 25 such distances as shown in Table 10. Paper [1] use displacement ratios calculated using the facial landmark instead of directly using the distances.



**Areas:**

We also calculated areas of the polygon surrounding the eye and the mouth region as shown in Figure 19. Paper [9] uses eccentricities of the ellipses instead of polygon areas. Area of polygon, whose co-ordinates are known is calculated using the formula:

|((𝑥1𝑦2 − 𝑦1𝑥2) + (𝑥2𝑦3 − 𝑦2𝑥3) + ⋯ (𝑥𝑛𝑦1 − 𝑦𝑛𝑥1))/2|

We calculate 3 such areas one for each eye and the mouth.

**Table 10****: Distances calculated on the face**

|  |  |
| --- | --- |
| **Distances** | **Facial points** |
| D1,D2,D3, D4,D5, D6, D7 and  D8 | Distances between the eyebrows and the upper eye-lid |
| D9,D10,D11 and D12 | Distances between the upper eye-lid and lower eye-lid |
| D13,D14,D15,D16 and D17 | Distance between the outer upper lip and outer lower lip |
| D18 and D25 | Distance between nose tip and outer upper lip and outer lower  Lip |
| D19 and D20 | Distance between nose tip and left and right mouth corner |
| D21 | Distance between the left and right mouth corner |
| D22 | Distance between inner points of both the eyebrows |
| D23 and D24 | Distance between nose tip and the inner points of both the  eyebrows. |

The prediction results using the 25 distance features and 3 areas (28 features) shows improvement in average accuracy by 2%. Also, along with SVM we use other two algorithms such as logistic regression and random forest classifier to evaluate our results.

Table 11 shows the result of experiment with 28 features consisting of distances and area. We have total 327 samples in a 70:30 split. Cross-validation folds are 5. Kernel for SVM is linear. For this experiment, Logistic Regression(l1 and l2 both) perform better along with Linear SVC(OVR). Logistic regression with penalty l1 gave the best result in terms of cross-validation score which has now improved to 82%..



**Figure 19: Polygon areas calculated on the face**

We use the cross\_val\_predict method from sklearn to predict the outcomes of all 327 sample files. The cross\_val\_predict method concatenates the result for each cross-validation result. As in our case, each data-point will be in the test set only once, this method gives the outputs for data points when they were present in the test set. The prediction for such a data point is not over-lapping and not averaged.

Prediction report Table 12 gives us results about how well the logistic regression algorithm did in terms of individual classification of classes for a 70:30 split of the data:

**10.INSTALLATION AND CODE IMPLEMENTAION**

**Step 1: Install Python:**

Visit the official Python website at https://www.python.org/

Download the latest version of Python suitable for your operating system.

Run the installer and follow the installation instructions.

Open a terminal or command prompt.

Install virtualenv (if not already installed):

pip install virtualenv

Create the environment for the project by using following command



Python -m venv raju (here the name raju is environment name)

Then activate the environment:

For windows verson

.\raju (environment name)\Scriots\activate

For unix or Linux

Source raju(environment name)/bin/activate

**Step 2: Install Required Libraries and Dependencies:**

Open a terminal or command prompt.

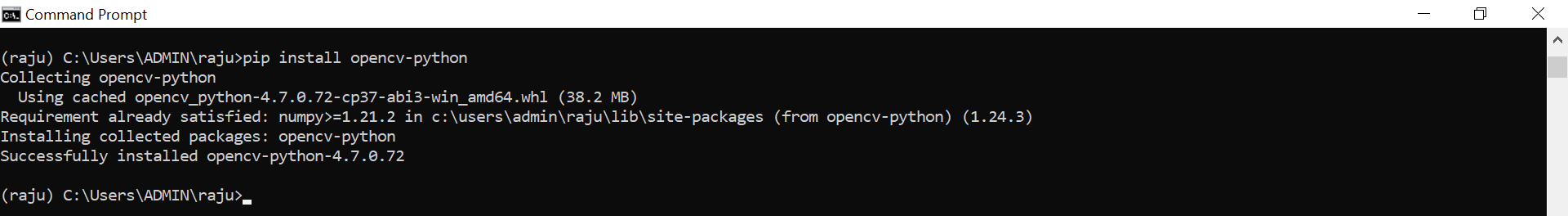
necessary libraries using the following commands Install the:

pip install package\_name

For OpenCV (computer vision library):

Note: Opencv supports only Python>=3.7 versions

There are two ways to install opencv library



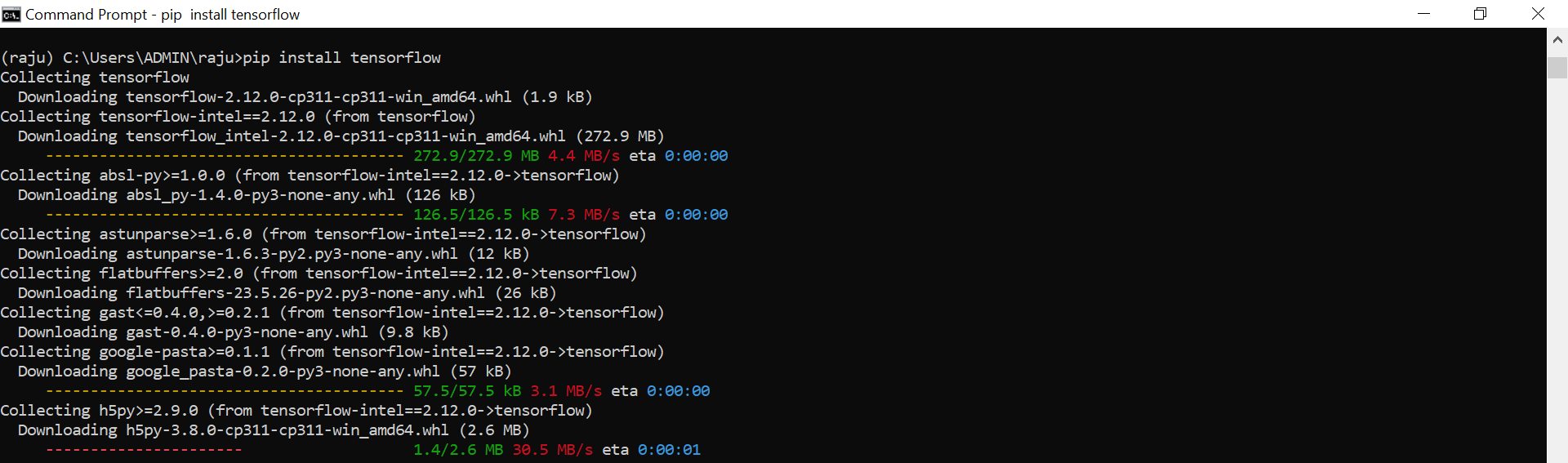
pip install OpenCV-python

This is an option to install with mail modules package

pip install OpenCV-contrib-python

This is for Full package (full package contains main modules and contrib/extra modules)

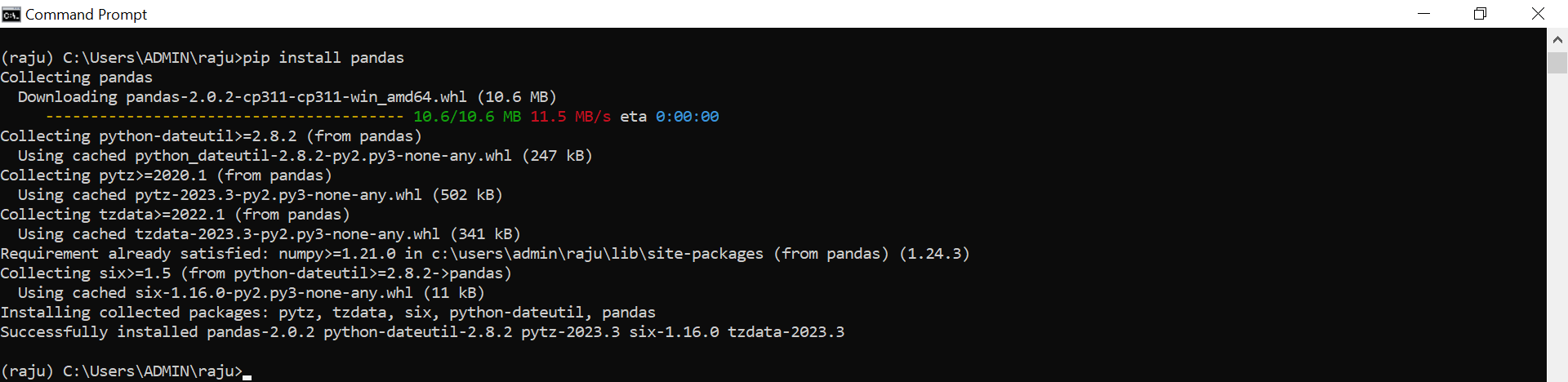
For TensorFlow (machine learning library):



pip install numpy



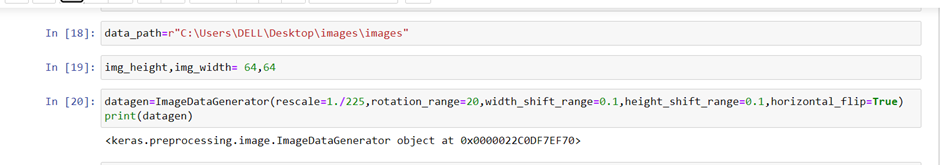
pip install pandas



**Step 3: Gather and prepare the input data:**

For emotions recognition, I gathered data in the form of images in jpeg format (Those size is 64, 64) i.e. img\_height , img\_width.

The following path is data path in my system



Here ImagedataGenerator is a function, it allows our model to receive new variations of the images at each epoch.

In this function I am using five arguments: Which are

**rescale:** rescaling factor. Defaults to None. If None or 0, no rescaling is applied, otherwise we multiply the data by the value provided (after applying all other transformations).

**rotation\_range:** Int. Degree range for random rotations

|  |  |
| --- | --- |
| **width\_shift\_range:** | Float, 1-D array-like or int   * float: fraction of total width, if < 1, or pixels if >= 1. * 1-D array-like: random elements from the array. * int: integer number of pixels from interval (-width\_shift\_range, +width\_shift\_range) - With width\_shift\_range=2 possible values are integers [-1, 0, +1], same as with width\_shift\_range=[-1, 0, +1], while with width\_shift\_range=1.0 possible values are floats in the interval [-1.0, +1.0). |
| **height\_shift\_range** | Float, 1-D array-like or int   * float: fraction of total height, if < 1, or pixels if >= 1. * 1-D array-like: random elements from the array. * int: integer number of pixels from interval (-height\_shift\_range, +height\_shift\_range) - With height\_shift\_range=2 possible values are integers [-1, 0, +1], same as with height\_shift\_range=[-1, 0, +1], while   with height\_shift\_range=1.0 possible values are floats in the interval [-1.0, +1.0 |

num\_classes=9

Here num-classes is variable name, and 9 represents the total number of emotion classes in my data. In this project, I am using 9 emotions classes. They are happiness, sadness, anger, surprise, disgust, wink, Gray pout, fear, neutral.

**4. Initializing/ Defining the Sequential model**

Initializing the Sequence model from Keras library. A sequential model is a stack of layers.

Here is the code for defining the sequential model.



model = keras.Sequential([

layers.Conv2D(32, (3, 3), activation="relu", input\_shape=(img\_height, img\_width, 3)),

In the above line I am initializing the Sequential model from keras library. Then I am adding the 2 Dimensional convolutional layer with 32 filters ,each filter size is 3X3, and activating the layer with RELU (Rectified liner unit) function, along with Input\_shape argument,

Input \_shape conations 4 arguments. They are

First parameter represents the Batch size, Second parameter represents the image height, third parameter represents the image width, and final argument represents the color channels (the input image is in RGB or BGR)

layers.MaxPooling2D((2, 2)),

After adding the convolutional layer I am adding the 2D MaxPooling with2x2 window layer for the reduce the spatial dimensions of the previous convolutional payer.

layers.Conv2D(64, (3, 3), activation="relu"),

Then I am adding the 2 Dimensional convolutional layer with 64 filters ,each filter size is 3X3, and activating the layer with RELU (Rectified liner unit) function. In this third layer I am not giving the input shape . Because model tekes the input only ince.

layers.MaxPooling2D((2, 2)),

After adding the convolutional layer I am adding the 2D MaxPooling with2x2 window layer for the reduce the spatial dimensions of the previous convolutional payer

layers.Flatten(),

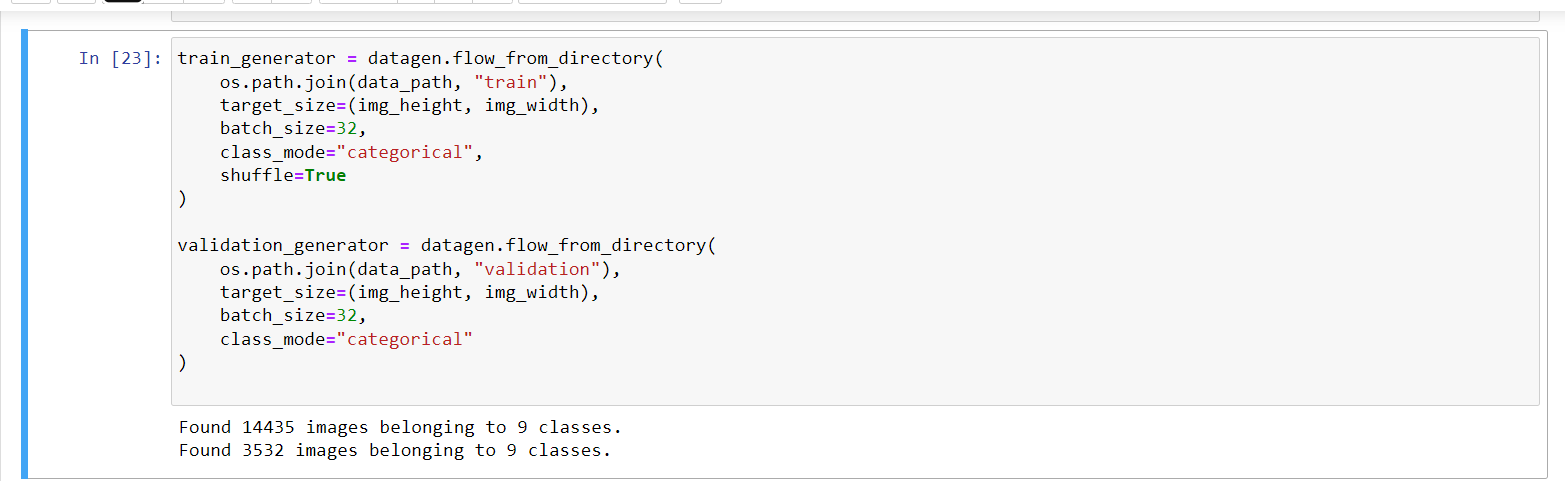
Here I am adding the flatten layer to the model. Flattening is used to convert the all results 2 Dimensional array from pooled function maps into a single long continuous liner vector.

layers.Dense(64, activation="relu"),

After flattening I am adding the fully connected layer to the model with 64 filters and applies the ReLU activation function

layers.Dense(num\_classes, activation="softmax")

This line adds the final fully connected layer with a number of units equal to the num\_classes variable. It uses the softmax activation function, which calculates the probabilities for each class.



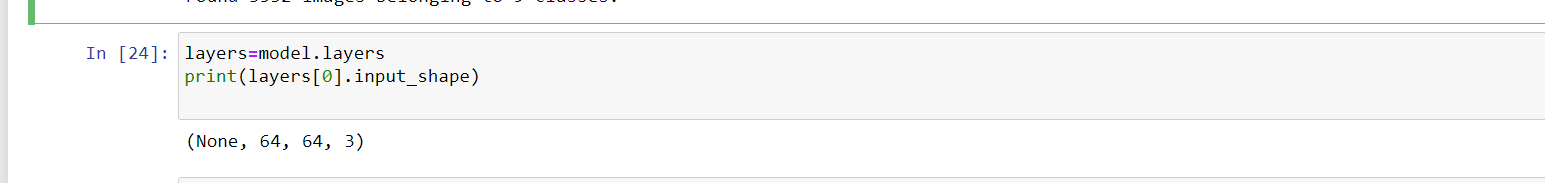
The function flow\_from\_directory method allows us to read the images directory from the directory and augment while the neural network model is learning on the data .

This function contains directory path argument, which represents path where my 9 classes of folders are present.

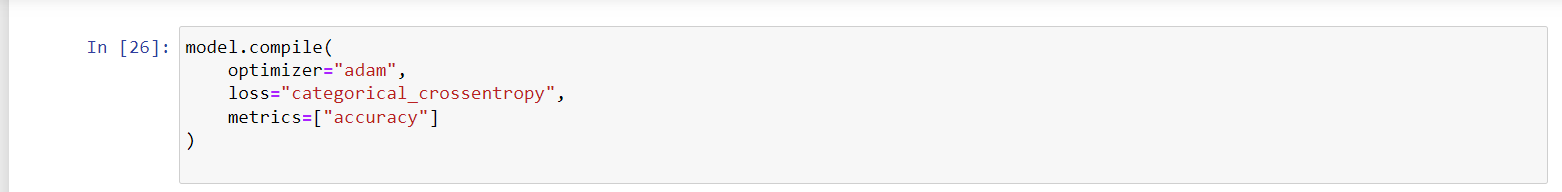
Target\_size argument represents, the size of my images, every image will be resized to this size.

Batch\_size argument represents, Number of images to be

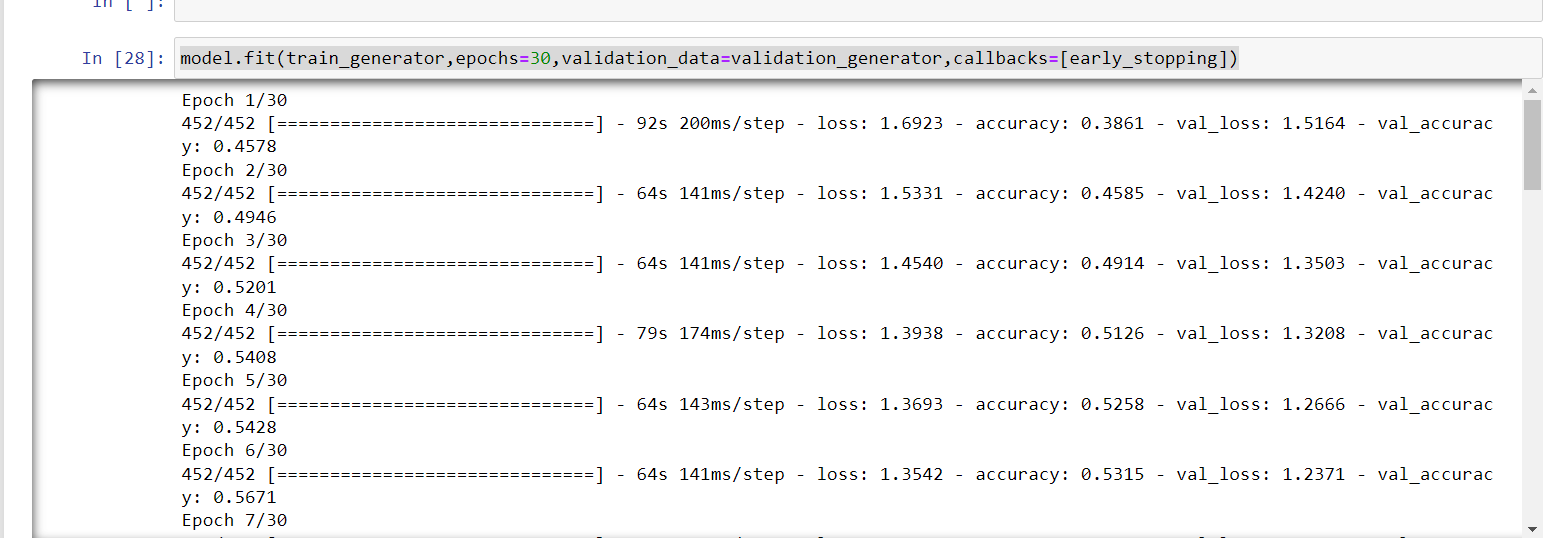
Note: This function also used for the validation images, here is only one thing is changed that is validation imagers path directory.



After building the model, I checked my model layers input shape and model summary with model.summary() and model.layers[0].input\_shape.



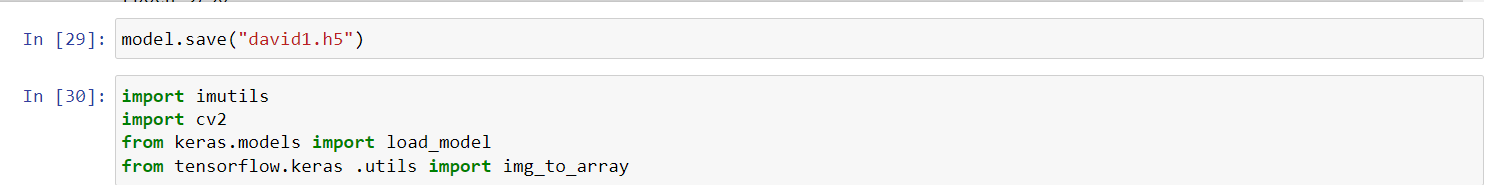
Model compilation is an activity performed after writing the statements in a model and before training starts. It checks for format errors, and defines the loss function, the optimizer or learning rate, and the metrics. A compiled model is needed for training but not necessary for predicting.



fit() is for training the model with the given inputs (and corresponding training labels). evaluate() is for evaluating the already trained model using the validation (or test) data and the corresponding labels. Returns the loss value and metrics values for the model. predict() is for the actual prediction.

According to Keras documentation, the model. fit method returns a history callback, which has a history attribute containing the lists of successive losses and other metrics.

In this project, I have taken 30 epochs for training the model

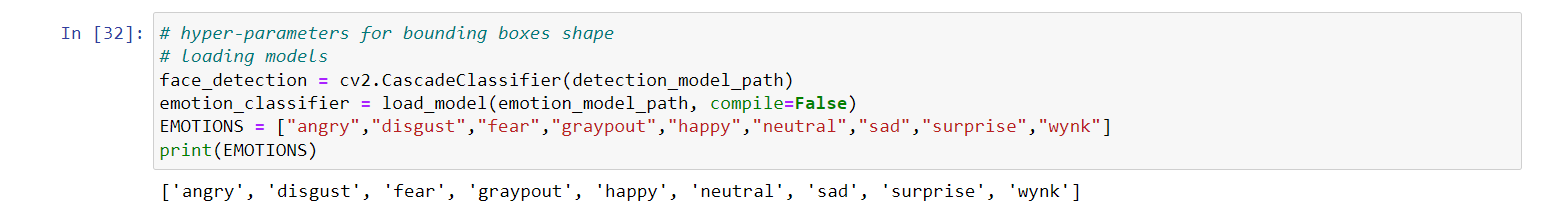


model.save() function saves the weights and the model structure to a single HDF5 file

**Step 4.5: Load pre-trained model:**

****

In this project, I am using ”haarcascade\_frontalface” for emotion detection. The Haar cascade classifier is a machine learning-based approach that uses Haar-like features to detect objects within images. It was introduced by Viola and Jones in their landmark paper in 2001, where they primarily focused on detecting faces.



The "haarcascade\_frontalface" algorithm is a specific implementation of the Haar cascade classifier that is trained to detect frontal faces in an image. It is particularly useful for face detection tasks, as it can quickly and accurately identify the presence and location of faces in images or video frames.

**11. CONCLUSIONS AND FUTURE WORK**

In this paper, a detailed analysis and comparison are presented on FER approaches. We categorized these approaches into two major groups: (1) conventional ML-based approaches and (2) DL-based approaches. The convention ML approach consists of face detection, feature extraction from detected faces and emotion classification based on extracted features. Several classification schemes are used in conventional ML for FER, consisting of random forest, AdaBoost, KNN and SVM. In contrast with DL-based FER methods, the dependency on face physics-based models is highly reduced. In addition, they reduce the preprocessing time to enable “end-to-end” learning in the input images. However, these methods consume more time in both the training and testing phases. Although a hybrid architecture demonstrates better performance, micro-expressions remain difficult tasks to solve due to other possible movements of the face that occur unwillingly.

Additionally, different datasets related to FER are elaborated for the new researchers in this area. For example, human facial emotions have been examined in a traditional database with 2D video sequences or 2D images. However, facial emotion recognition based on 2D data is unable to handle large variations in pose and subtle facial behaviors. Therefore, recently, 3D facial emotion datasets have been considered to provide better results. Moreover, different FER approaches and standard evaluation metrics have been used for comparison purposes, e.g., accuracy, precision, recall, etc.

FER performance has increased due to the combination of DL approaches. In this modern age, the production of sensible machines is very significant, recognizing the facial emotions of different individuals and performing actions accordingly. It has been suggested that emotion-oriented DL approaches can be designed and fused with IoT sensors. In this case, it is predicted that this will increase FER’s performance to the same level as human beings, which will be very helpful in healthcare, investigation, security and surveillance.