UROP Report

Submitted

By

Kolli Sukhdev AP19110010276

Nallapaneni Sai Suprabhanu AP19110010232

Gopu Sai Alekhhya AP19110010235

Valeti Manish AP19110010204

Department of Computer Science and Engineering

SRM University-AP, Andhra Pradesh, India

April - 2022

DATA SHEET

Roll Numbers	:	1. AP191110010276 2. AP191110010235 3. AP191110010232 4. AP191110010204
Name of the student	:	1. Kolli Sukhdev 2. Gopu Sai Alekhhya 3. Nallapaneni Sai Suprabhanu 4. Valeti Manish
Title of the Project	:	Integration of student gestures and facial emotion recognition to virtual meeting platforms along with the statistics of their concentration level during the class
Branch & Section	:	CSE-B
Batch	:	2019-2023
Start Date (MM/DD/YYYY)	:	01-02-2022
End Date (MM/DD/YYYY)	:	30-04-2022
Status of the project	:	Completed
Name of UROP mentor (SRM Faculty)	:	Dr. Ravi Kant Kumar

TABLE OF CONTENTS

ACKNOWLEDGEMENT

We would like to convey our gratitude in particular to our mentor “Dr. Ravi Kant Kumar” for his patience, guidance and support in teaching the course and helping us to complete the project by clarifying all the doubts we got while completing it. The way our professor taught us the topics are never forgettable and easy to learn in a short period of time. This project will be a big achievement that will improve our confidence and discipline for completing any work on time and using it in further conditions. We also thank each of our teammates for helping and encouraging each other to complete the project in a better way.

Abstract:

Facial recognition has become one of the most researched and sought-out topics in recent years. There is a huge amount of research work that is being done around the world in the same area. It is used in many applications such as camera surveillance. Emotion detection is a subset of facial recognition which focuses mainly on the emotions that a human face is expressing. The main aim of this paper is to integrate the emotion detection methods into virtual meeting platforms like zoom and google meet where we can get the mood of the class. The model we developed helps in getting the overall statistics of the mood and emotion that the participants or students attending the meet are in. Also one of the main important differences between our model and various other models is that it can also work well under low light conditions. While keeping the main objective of the paper to focus on real-time emotion detection in virtual meeting platforms such as Zoom, Google meets, etc; the paper also focuses on many aspects to improve the virtual meeting platform experience.

1. Introduction:

While so much research is taking place in the field of computer vision, there is still so much more where OpenCV can be applied and studied. Recognizing emotion detection can sometimes be hard even for the human eye but by using machine learning algorithms we can identify the emotion on a person's face with much more accuracy.

The fundamental reasoning for this decision is to provide the user with a more comprehensive assessment: because emotions can only be comprehended in the context of a person's own qualities, we reasoned that examining personality features would provide a new key to understanding emotional variations. Our ultimate goal is to improve the user experience and the quality of our analysis by incorporating any relevant and supplementary data that will help us better understand the user's quirks.

The model we have developed is of great use to virtual meeting platforms such as Zoom and Google Meet, where the host can get to know the overall mood of the class or during interviews where candidates' emotions can be studied are some of the applications of this model. This can also work well in low-light conditions. There are many other facial detection algorithms and models previously developed but this is an application that will be integrated with virtual meeting platforms that can also work well under low light conditions. The future scope of this project is using hand gestures to zoom in and out of the presentation screen. Future work also includes functions related to the memory aspect of Zoom platforms where proctoring can be made easy by creating a buffer for a certain time limit to save memory and reduce the bandwidth required during live proctoring.

Emotion has been examined and investigated for many years and has been shown to be difficult to categorize. When most individuals think of emotion, they think of it. They consider their own feelings. There are many different types of psychology. Happiness, anger, sadness, contempt, surprise, and fear are six recognized archetypal emotions. While these feelings are there, True, not many people think about what they're saying or how we as individuals are able to perceive and respond to them. Most of the time, we have the capacity to see someone in the eyes and tell what emotion they are feeling based on their facial expressions

2. Methodology:

The main goal of this project is to detect the emotion of the person during live meeting sessions. This can be used for many applications such as live interviews, and online classes where the host wants to know the report of the total emotion of the class.

- Video input is taken or real-time video is taken as the input and the emotion on the person's face is analyzed.
- The statistics of the various emotions will also be displayed visually with the help of a bar graph.

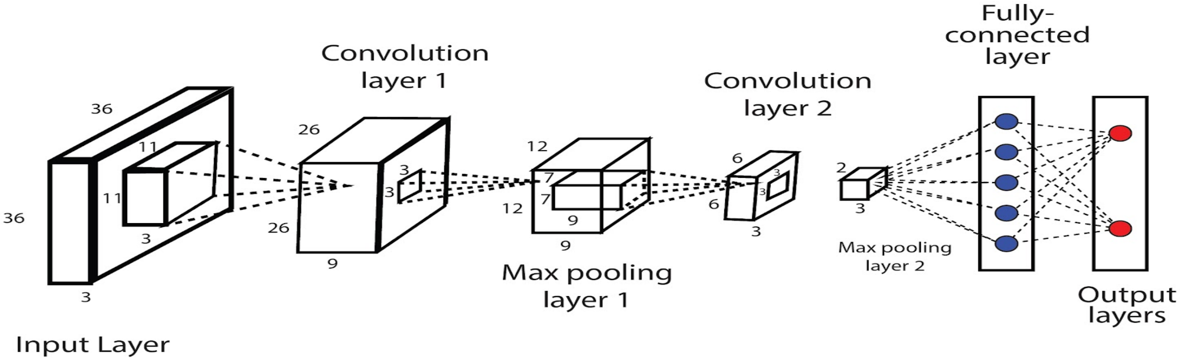
3. Related Work:

Previous work on emotion recognition does already exist but its performance has been significantly improved. The model is also different in terms of low light performance and compatibility with low clarity cameras. The model was trained on the largest publicly available datasets. This made it possible to improve the recognition efficiency. Integration of this is not applied as of date.

The memory buffer application would also help improve the performance in terms of memory. The datasets that are available would only be indicative and may differ sometimes in the real world. But these limitations only apply to this and would not impact the performance of the model.

4. Architecture:

Convolutional Neural Network



The new era requires new standards and the Convolutional Neural Networks (CNNs) are the unique types of neural networks for processing the data with a grid-like topology.

In traditional SVM approaches, selecting the filters like Gabor filters and the architecture of the filters was a large part of the job in order to extract as much information from the image as feasible. With the rise of deep learning and machine learning and with increased computing power, this work can now be automated. The name of CNNs is derived from the fact that the original picture input is convolved with a collection of filters. The number of filters to apply remains the parameter to choose as well as the size of the filters. The stride length refers to the filter's dimension. Stride lengths typically range from 2 to 5.

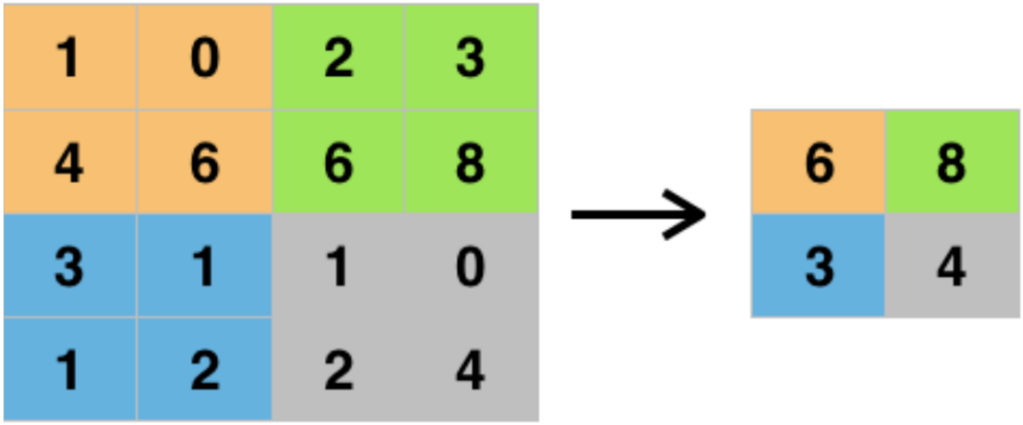
In some ways, we're creating a volume convolved output. It's no longer a two-dimensional representation. The filters are difficult to comprehend, especially when we utilize a large number of them. Some are used to locate curves, edges, and textures, among other things... After extracting the volume, it can be flattened and fed into a dense Neural Network. The convolution is stated mathematically as:

$$(f * g)(t) = \int_0^t f(\tau)g(t-\tau)\partial\tau$$

The convolution represents the percentage of the area of the filter g that overlaps with the input f at time τ overall time t. Because $\tau < 0$ and $\tau > t$ are meaningless, the convolution can be reduced to:

$$(f * g)(t) = \int_{-\infty}^{+\infty} f(\tau)g(t-\tau)\partial\tau$$

We apply an activation function to each input at each convolution stage (ReLU). We've simply added dimensionality to our original image input so far. A pooling process is then used to minimize the dimension. Pooling entails downsampling features to reduce the number of parameters that must be learned during training. Max-pooling is the most prevalent type of pooling. We execute a max-pooling for each of the input image's dimensions, which takes the maximum value among the 4 pixels over a specific height and width, commonly 2x2. When classifying a picture, the intuition is that the maximal value has a higher likelihood of being more relevant.



1	0	2	3
4	6	6	8
3	1	1	0
1	2	2	4

6	8
3	4

All the components of CNN are now covered:

- Convolution layer
- Activation
- Pooling layer
- Fully connected layer, similar to a dense neural network

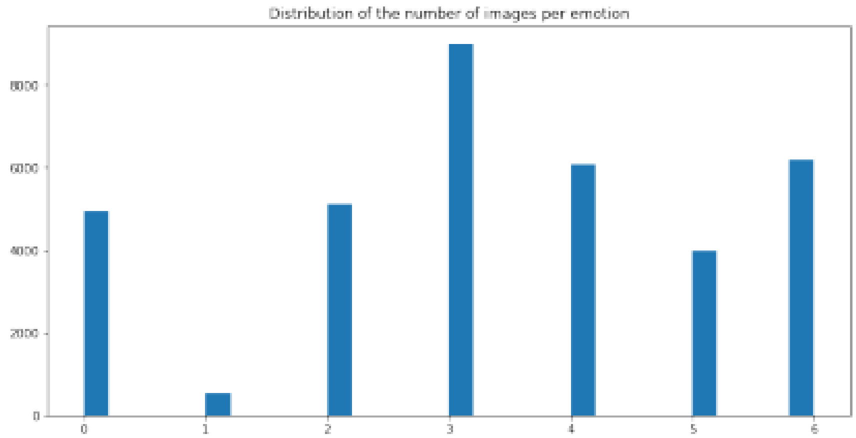
The order of the layers can be switched:

$$\mathbf{ReLU(MaxPool(Conv(X))) = MaxPool(ReLU(Conv(X)))}$$

We commonly use several layers of convolution and pooling to classify images. We can now model more complex structures. The majority of deep learning model tuning involves determining the best model structure. Some well-known algorithms created by Microsoft or Google include over 150 hidden layers.

5. Implementation:

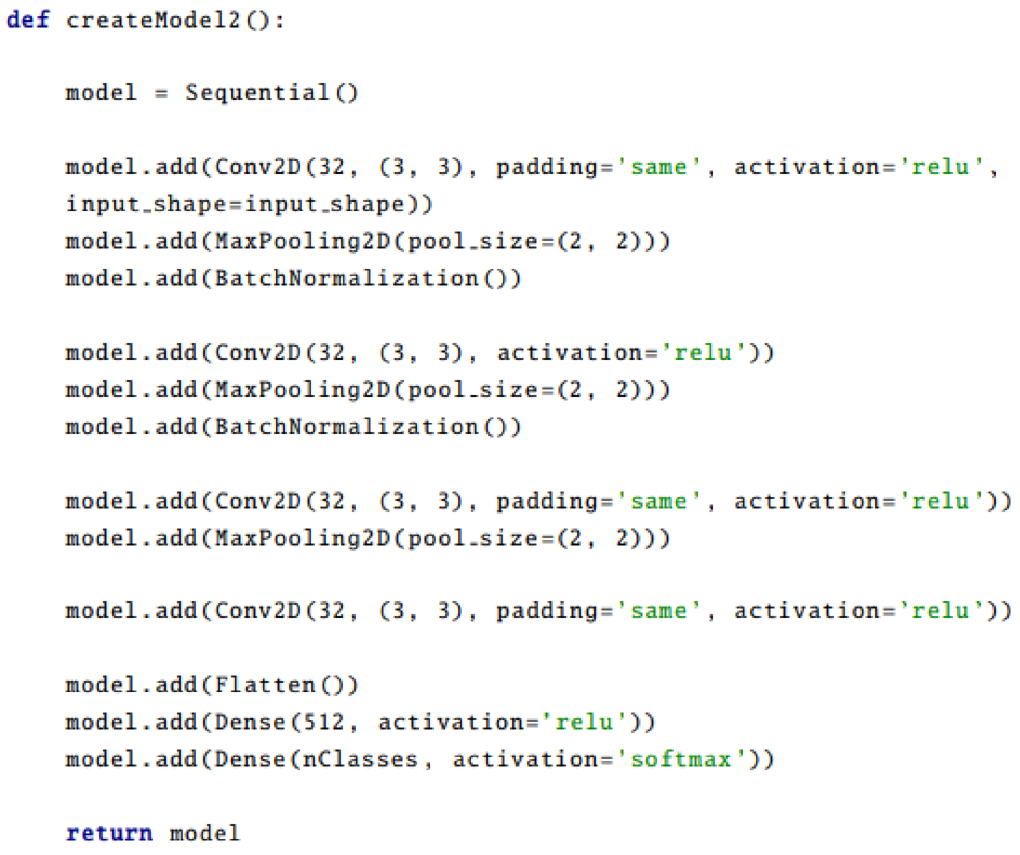
First of all, when exploring the data of the FER2013 data set, we observe that there is an imbalance in the number of images by class (emotion). The labels are the following :



0: Angry
1: Disgust
2: Fear
3: Happy
4: Sad
5: Surprise
6: Neutral

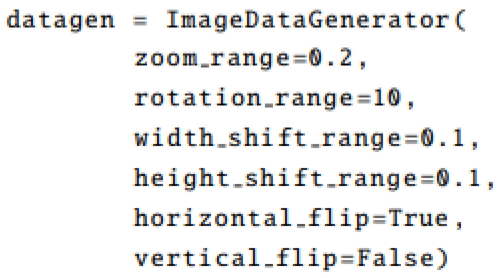
We will therefore need to explore data augmentation techniques to solve this issue. The train set has 28709 images, the test set has 3589 images. For each image, the data set contains the grayscale color of 2304 pixels (48x48), as well as the emotion associated. We achieved an accuracy of over 80% even in low light conditions and with low-resolution cameras. There is a little bit of overfitting in the model while estimating disgust, certainly caused due to the low number of images of disgust emotion relative to other emotions.

Not surprisingly, deep learning solutions perform better than more classical SVM algorithms in the literature. The deep learning architecture we will be using is based on the best-performing algorithm from the paper cited above. The corresponding architecture in Keras is the following.



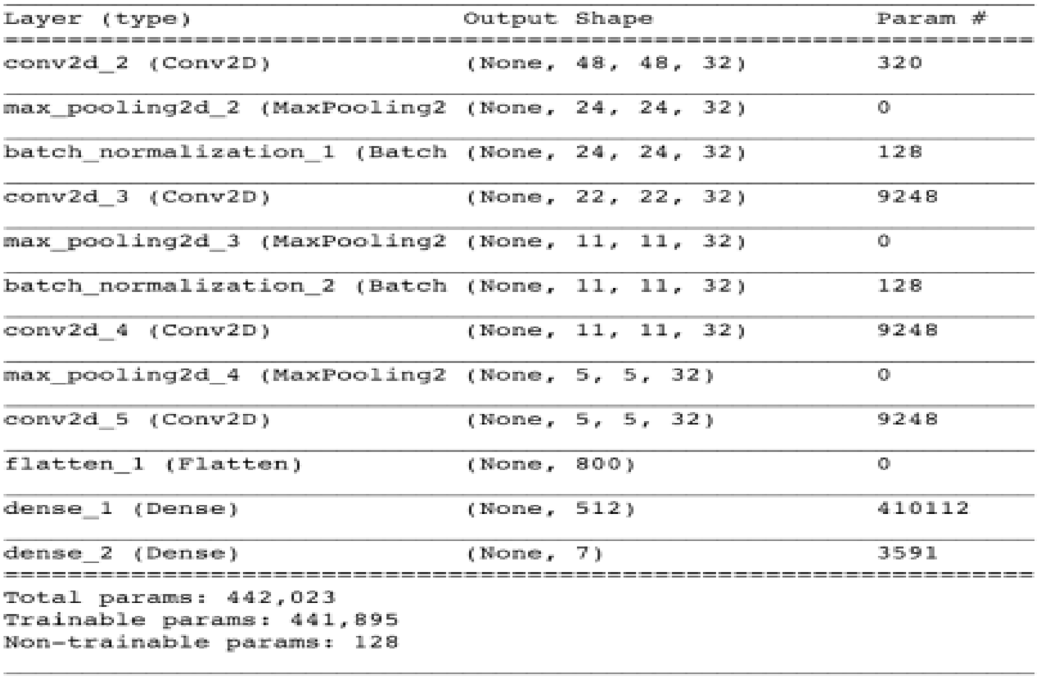
```
def createModel2():

    model = Sequential()

    model.add(Conv2D(32, (3, 3), padding='same', activation='relu',
    input_shape=input_shape))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(BatchNormalization())

    model.add(Conv2D(32, (3, 3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(BatchNormalization())

    model.add(Conv2D(32, (3, 3), padding='same', activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(32, (3, 3), padding='same', activation='relu'))

    model.add(Flatten())
    model.add(Dense(512, activation='relu'))
    model.add(Dense(nClasses, activation='softmax'))

    return model
```

This simple architecture produces over 440’000 parameters to estimate. The computation time is around 8 hours on the local machine. In order to prevent overfitting, we also apply Keras built-in data generation module.

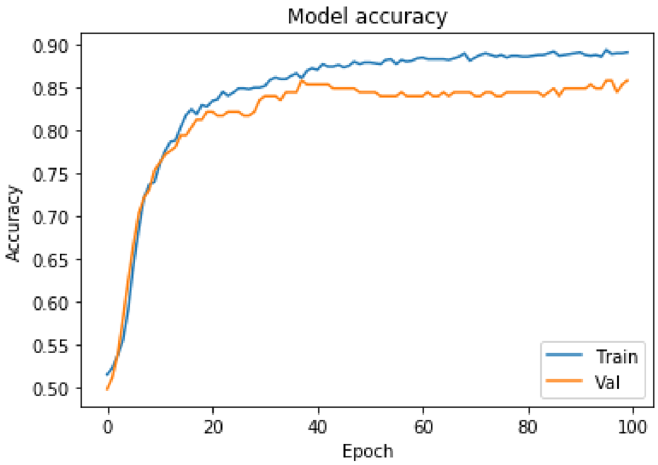


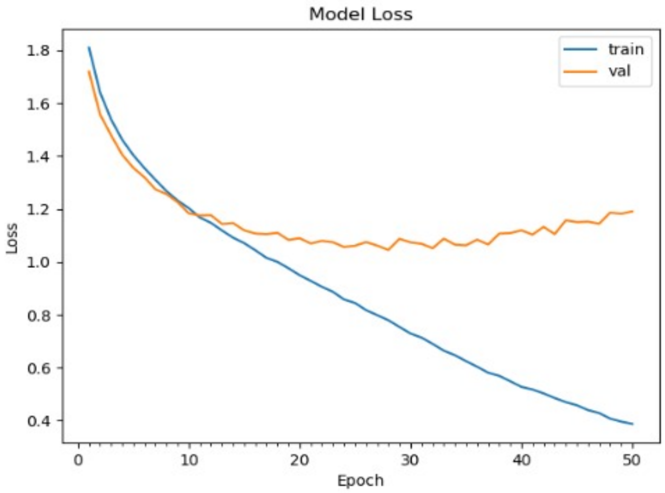
```
datagen = ImageDataGenerator(
        zoom_range=0.2,
        rotation_range=10,
        width_shift_range=0.1,
        height_shift_range=0.1,
        horizontal_flip=True,
        vertical_flip=False)
```

The optimizer we chose is RMSprop, an optimizer that divides the learning rate by an exponentially decaying average of squared gradients. The loss we use is the categorical cross-entropy since we face a classification problem. Finally, the metric we use is accuracy.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_2 (Conv2D)            (None, 48, 48, 32)        320
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 24, 24, 32)        0
_________________________________________________________________
batch_normalization_1 (Batch (None, 24, 24, 32)        128
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 22, 22, 32)        9248
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 11, 11, 32)        0
_________________________________________________________________
batch_normalization_2 (Batch (None, 11, 11, 32)        128
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 11, 11, 32)        9248
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 5, 5, 32)          0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 5, 5, 32)          9248
_________________________________________________________________
flatten_1 (Flatten)          (None, 800)               0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               410112
_________________________________________________________________
dense_2 (Dense)              (None, 7)                 3591
=================================================================
Total params: 442,023
Trainable params: 441,895
Non-trainable params: 128
_________________________________________________________________
```





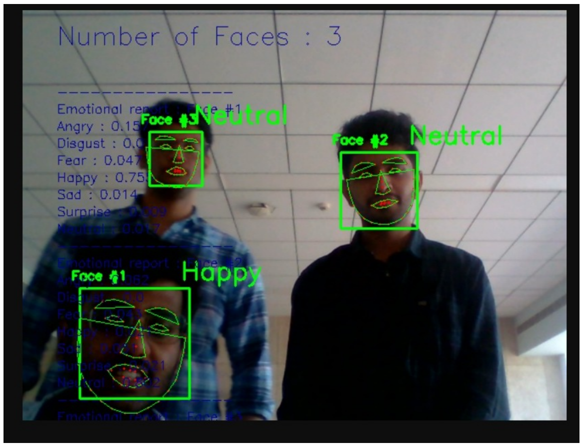
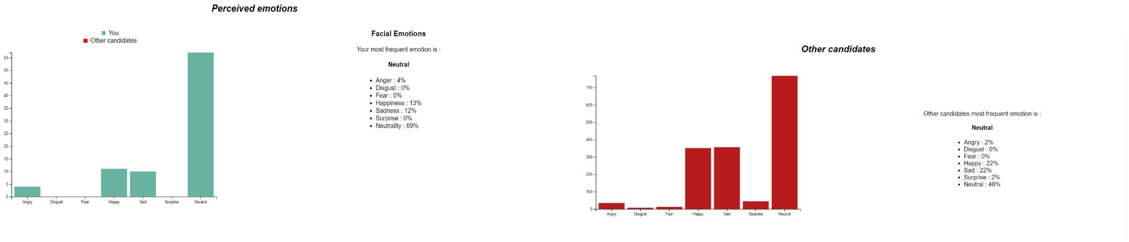


Illustration of the working model



Facial Emotions

Your most frequent emotion is :

Neutral

- Anger : 4%
- Disgust : 0%
- Fear : 0%
- Happiness : 13%
- Sadness : 12%
- Surprise : 0%
- Neutrality : 69%

Other candidates most frequent emotion is :

Neutral

- Angry : 2%
- Disgust : 0%
- Fear : 0%
- Happy : 22%
- Sad : 22%
- Surprise : 2%
- Neutral : 48%

Statistics of other's emotions

6. Scope of the Project:

While the current implementation and execution of the project are limited to recognizing whether a student is present during the class or not and calculating the emotion that is captured on the face, the future implementation to improve the virtual meetings would be adding a feature that would help in optimizing the aspects that are related to the performance of application such as memory management and bit transfer rate.

The project would expand to various applications such as using the feature to calculate the mood of a candidate in an interview. This would help in improving the interview experience in virtual mode. The scope of the project would also expand to even low-memory devices.

7. Future Direction:

Going forward more research and more information about facial recognition and emotion detection will be looked into. Now that emotions are being successfully detected, the next step is to detect more emotions with more accuracy.
The future scope of the project would expand to a project where a memory buffer will be created with a fixed or variable time period wherein the video that is being recorded and stored in the memory will be deleted if no disturbance is noticed.

There has been a lot of study on face feature extraction and feature classification in recent years (emotion detection). Zhang et al., for example, use a multimodal method for facial emotion recognition. They concentrated their study on two sorts of modalities: texture and landmark-based. Then, for feature classification, a support vector machine (SVM) is trained, which is a common choice for categorizing features. SVMs have supervised learning modules (they are provided training data) that contain classification and regression analysis methods that may subsequently be utilized to generate a hyperplane.

8. Conclusion:

To properly express emotion from photos, first extract the required face characteristics, which are then applied to various action units (AUs) or straight to the classifiers. When it comes to portraying particular emotions, facial mobility is crucial. The muscles of the face are deliberately manipulated to represent particular emotions that humans can identify, even if the expression is very subtle.

Finding the geometrics of particular face locations is one way of detecting mood. This enables the general form and placement of each face feature to be properly determined.

Determining whether or not a face is present is perhaps the most important step in the facial recognition process. Mainly because it is essentially the beginning of the first step in the process (besides preparing the data set). OpenCV, the software used for facial recognition in this project, uses classifiers to detect objects. These classifiers are trained by the programmer for whatever it is that he or she wants the recognition software to detect.

9. References:

1. Tzirakis, P., Trigeorgis, G., Nicolaou, M. A., Schuller, B. W., & Zafeiriou, S. (2017). End-to-end multimodal emotion recognition using deep neural networks. *IEEE Journal of Selected Topics in Signal Processing*, *11*(8), 1301-1309
2. Pramerdorfer, C., & Kampel, M. (2016). Facial expression recognition using convolutional neural networks: state of the art. *arXiv preprint arXiv:1612.02903*.
3. Davis, J., & Shah, M. (1994). Visual gesture recognition. *IEE Proceedings-Vision, Image and Signal Processing*, *141*(2), 101-106
4. Ng, C. W., & Ranganath, S. (2002). Real-time gesture recognition system and application. *Image and Vision computing*, *20*(13-14), 993-1007.
5. Itoh, M., & Chua, L. O. (2008). Imitation of visual illusions via OpenCV and CNN. *International Journal of Bifurcation and Chaos*, *18*(12), 3551-3609.
6. Kandjimi, H., Srivastava, A., & Muyingi, H. (2021, August). Attendance System with Emotion Detection: A case study with CNN and OpenCV. In *Proceedings of the International Conference on Data Science, Machine Learning and Artificial Intelligence* (pp. 53-56).
7. The Facial Emotion Recognition Challenge from Kaggle, https://www.kaggle.com/c/challenges-in-representation-learning-facial-expressionrecognition-challenge/data (Available).
8. Dataset taken FER2013