Emojify videos in real time

Team Members:

Suryavijoy Kar- 190020039

Kushagra Khatwani - 190020021

Prateek Chawla- 190020032

Jyoti Shukla- 211022003

Overview

In this project we will be working on videos of facial expressions to decode the emotions represented in it, that is to classify human facial expressions. Using the emotions classified then we will try to map those facial expressions into an emoji.

Goals

1. Facial Expression Recognition from live video
2. Conversion of facial expressions into emoticons in real time

Approach

1. Look into various preprocessing techniques and prepare the data set.
2. Detecting faces and creating bounding boxes in real time
3. Detecting and predicting emotions on live faces.
4. Emojify obtained facial emotions in real time.
5. (Tentative) Audio modality inclusion in our project.

Research Papers review

Some relevant research papers following different strategies for facial recognition and emotion recognition are briefed below:

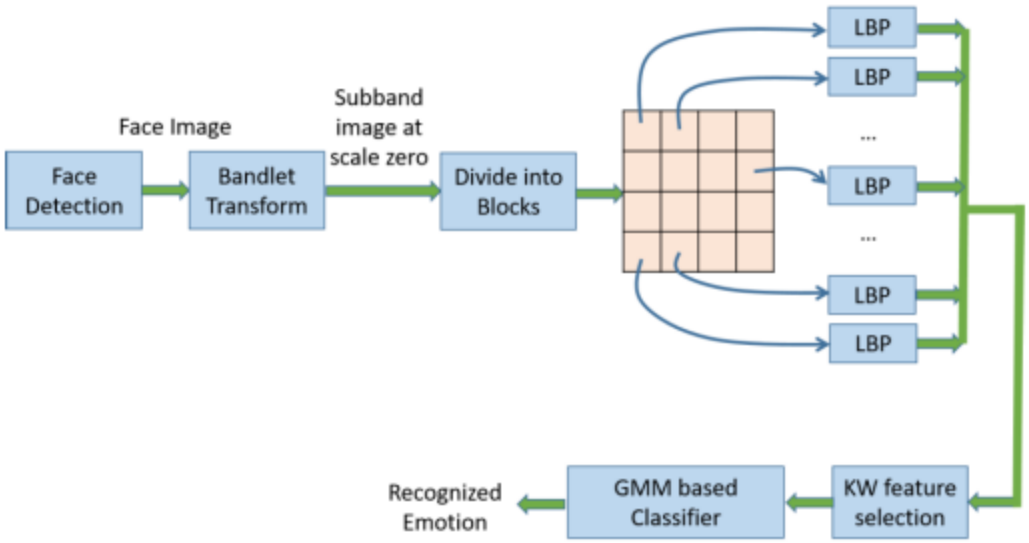
1. **An Emotion Recognition System for Mobile Applications (Hussain and Muhammad 2017, 7):**

   In this paper, they proposed an emotion recognition system suitable for mobile applications which is computational efficient, requires less storage and processing

time. In the proposed system, facial video is captured by an embedded camera of a smartphone. Some representative frames are extracted from the video, and a face detection module is applied to extract the face regions in the frames.

The Bandlet transform is realized on the face regions,and the resultant sub-band is divided into non-overlapping blocks. Local binary patterns' histograms are calculated for each block, and then are concatenated over all the blocks.

The Kruskal–Wallis feature selection is applied to select the most dominant bins of the concatenated histograms. The dominant bins are then fed into a Gaussian mixture model-based classifier to classify the emotion.



Block diagram of the proposed emotion recognition system.

The frame selection from the video will be done by calculating the chi-squared distance between the histograms of two successive gray-scaled frames. The face areas in the frames are detected by looking into various preprocessing techniques and prepare the data set. To overcome the obstacle of representing geometric structure in less time, bandlet transform is used. The next step is to apply LBP on the subband images of the bandlet transform. LBP histograms are calculated and concatenated to describe the feature set for the image. To sort features from this huge feature set KW feature selection method is used. The selected features are fed to the GMM based classifier. GMM classifier can work in real time. They used two large databases for training and testing their model. Considerable good results were achieved.

2. Rapid Object Detection using a Boosted Cascade of Simple Features:

This paper focuses on the machine learning approach for visual object detection which proves to have high accuracy and high detection rates.

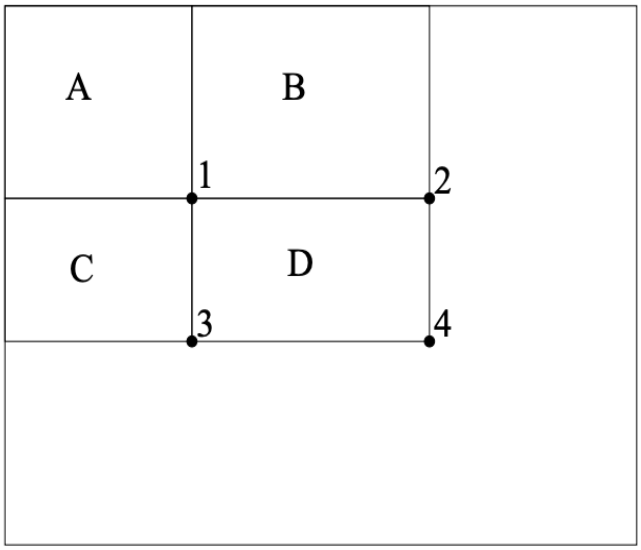
This was achieved by using:

- Introduction of a new image representation called the “Integral Image” which allows the features used by the model to be computed very quickly.
- The second is a learning algorithm, based on AdaBoost, which selects a small number of critical visual features from a larger set and yields extremely efficient classifiers.
- The third contribution is a method for combining increasingly more complex classifiers in a “cascade” which allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions.

Integral Image:

Rectangle features can be computed rapidly using an intermediate representation for the image.

The integral image at a location x,y consists of the sum of pixel values above and left of x,y.



Usually depending on the window size i.e rectangle dimensions considered this will lead to a large number of rectangular features which can be easily calculated in one pass using recursion.

Adaboost algorithm used:

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:

  1. Normalize the weights,

  $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$

  so that w_t is a probability distribution.

  2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|$.
  3. Choose the classifier, h_t, with the lowest error ϵ_t.
  4. Update the weights:

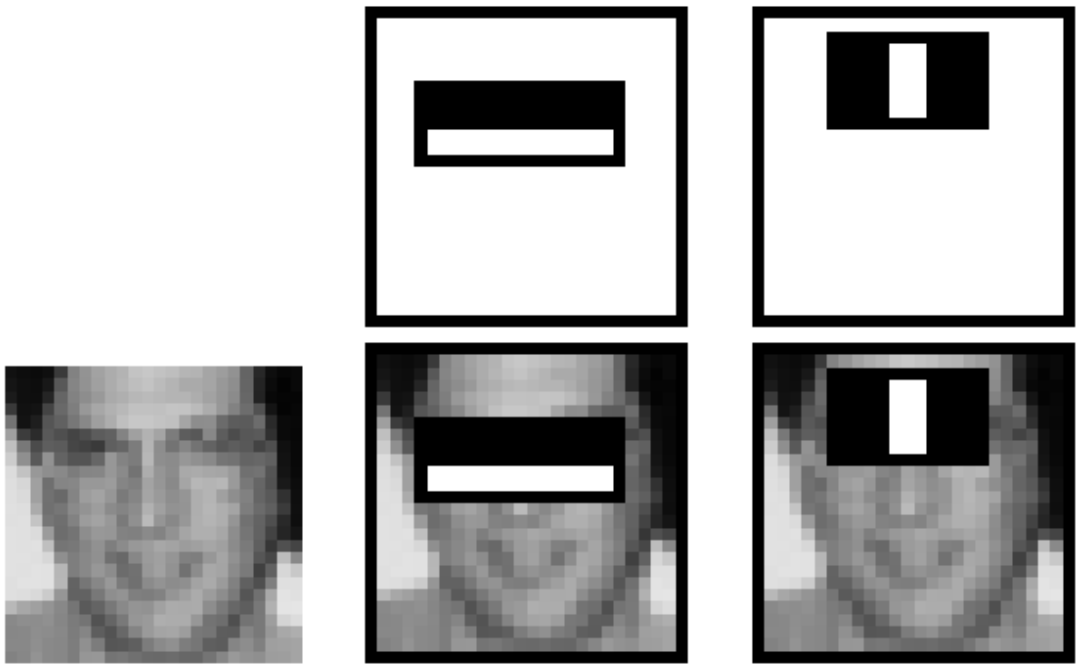
  $$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

  where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

- The final strong classifier is:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

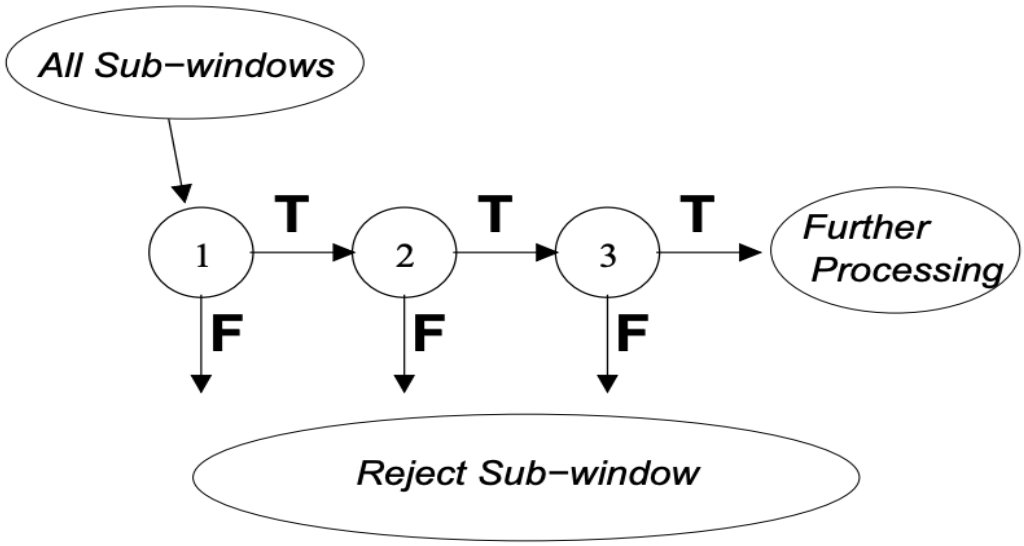
where $\alpha_t = \log \frac{1}{\beta_t}$



The first and second features selected by AdaBoost. The two features are shown in the top row and then overlaid on a typical training face in the bottom row. The first feature measures the difference in intensity between the region of the eyes and a region across the upper cheeks.

The feature capitalizes on the observation that the eye region is often darker than the cheeks. The second feature compares the intensities in the eye regions to the intensity across the bridge of the nose.

Cascade of classifiers:



Schematic depiction of the detection cascade. A series of classifiers are applied to every sub-window. The initial classifier eliminates a large number of negative examples with very little processing. Subsequent layers eliminate additional negatives but require additional computation. After several stages of processing the number of sub-windows have been reduced radically. Further processing can take any form such as additional stages of the cascade

References

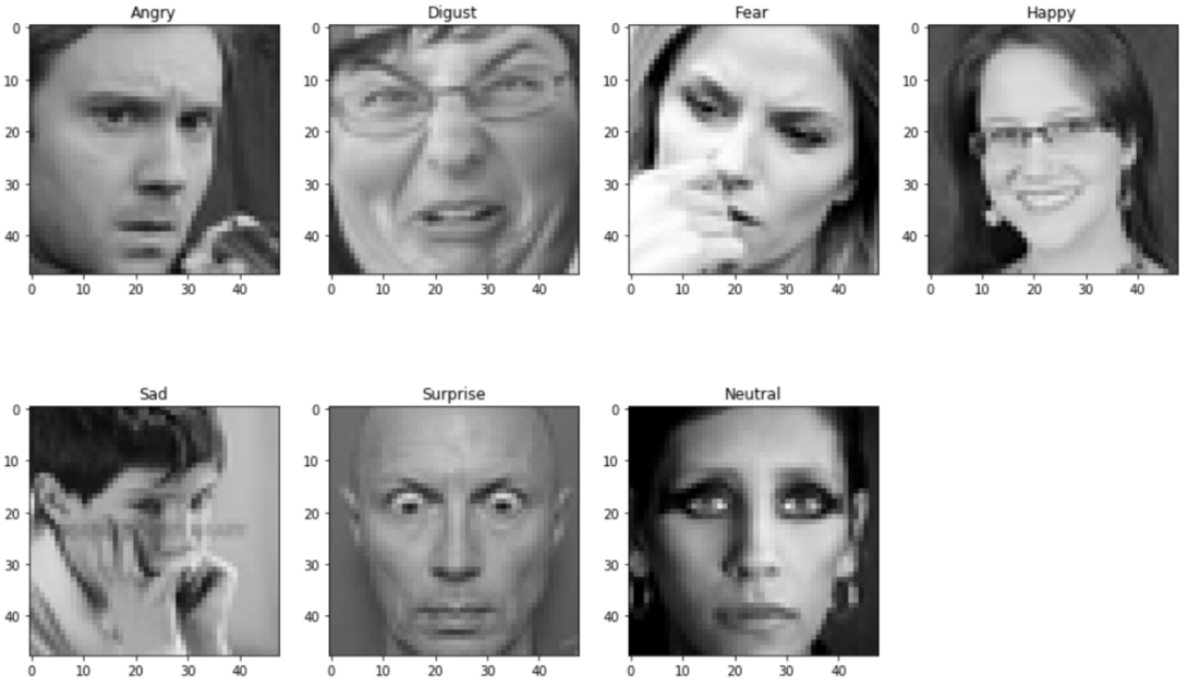
1. Hussain, M. Shamim, and Ghulam Muhammad. 2017. “An Emotion Recognition System for Mobile Applications. *IEEE Access* 5:2281-2287. https://ieeexplore.ieee.org/document/7862118.
2. Paul Viola, Michael Jones ,2001 Rapid Object Detection using a Boosted Cascade of Simple Features

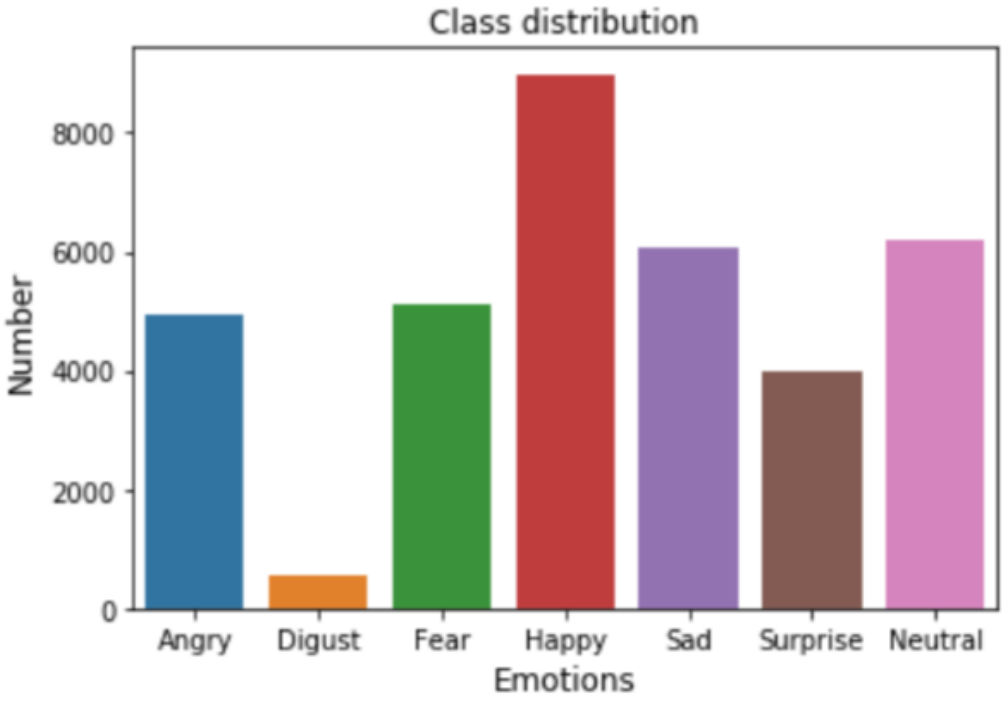
Datasets

We will be using one of the open resource datasets available at Kaggle. This dataset includes 35,887 unique images stored in a format of pixel values with corresponding class labels. There are seven facial emotion categories: **anger**, **disgust**, **fear**, **happiness**, **sadness**, **surprise**, and **neutral**.

	emotion	pixels
0	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...
1	0	151 150 147 155 148 133 111 140 170 174 182 15...
2	2	231 212 156 164 174 138 161 173 182 200 106 38...
3	4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...
4	6	4 0 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...

	emotion	number
0	Angry	4953
1	Digust	547
2	Fear	5121
3	Happy	8989
4	Sad	6077
5	Surprise	4002
6	Neutral	6198


Angry
Digust
Fear
Happy
Sad
Surprise
Neutral


Class distribution
Number
8000
6000
4000
2000
0
Angry
Digust
Fear
Happy
Sad
Surprise
Neutral
Emotions

(For including Audio modality)

The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)

Multimodal database of emotional speech and song. The database is gender balanced consisting of 24 professional actors, vocalizing lexically-matched statements in a neutral North American accent. Speech includes calm, happy, sad, angry, fearful, surprise, and disgust expressions, and the song contains calm, happy, sad, angry, and fearful emotions. All conditions are available in face-and-voice, face-only, and voice-only formats.

Meet 2

Recap:

- Explained goal and approach
- Discussed 2 research papers involving emotion recognition and object detection (cv2 library)
- Dataset

Dataset:

- Current dataset for training CNN model : FER 2013 Dataset (Kaggle)
- Video dataset used for testing emotion detection model : tracking-shot-of-excited-people

Model:

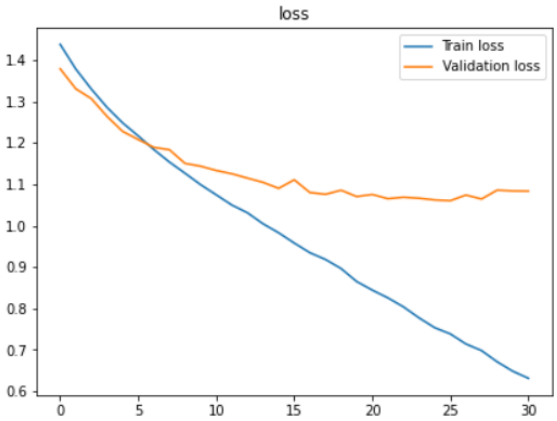
Bounding boxes:

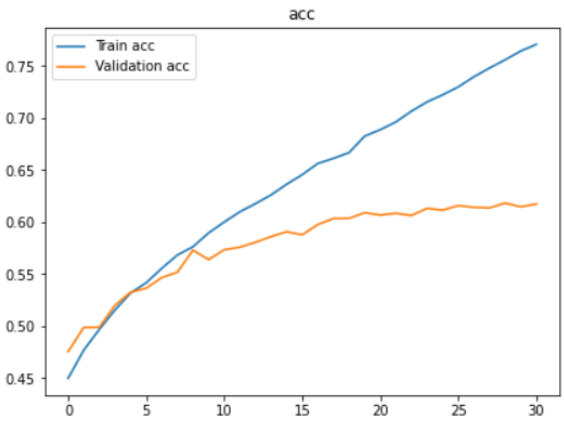
- Object Detection using Haar feature-based cascade classifiers.
- Even a 24x24 window results in over 160000 features.So, internal image introduced.
- Most of the image is non-face region, so a cascade of classifiers is used.
- According to the authors, on average 10 features out of 6000+ are evaluated per sub-window.

Summary of CNN model:

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 46, 46, 32)        320
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 44, 44, 64)        18496
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 22, 22, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 22, 22, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 20, 20, 128)       73856
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 10, 10, 128)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 8, 8, 128)         147584
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 4, 4, 128)         0
_________________________________________________________________
dropout_1 (Dropout)          (None, 4, 4, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 1024)              2098176
_________________________________________________________________
dropout_2 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_1 (Dense)              (None, 7)                 7175
=================================================================
Total params: 2,345,607
Trainable params: 2,345,607
Non-trainable params: 0
_________________________________________________________________
```

Results:





Training Loss	0.5
Training accuracy	78%
Validation Loss	1.1
Validation accuracy	62%

- Result on pre-recorded video

Future challenges:

- Improve model performance
- Improve bounding boxes detection performance
- Extend emotion detection from pre-recorded videos to live videos

Meet 3

Recap

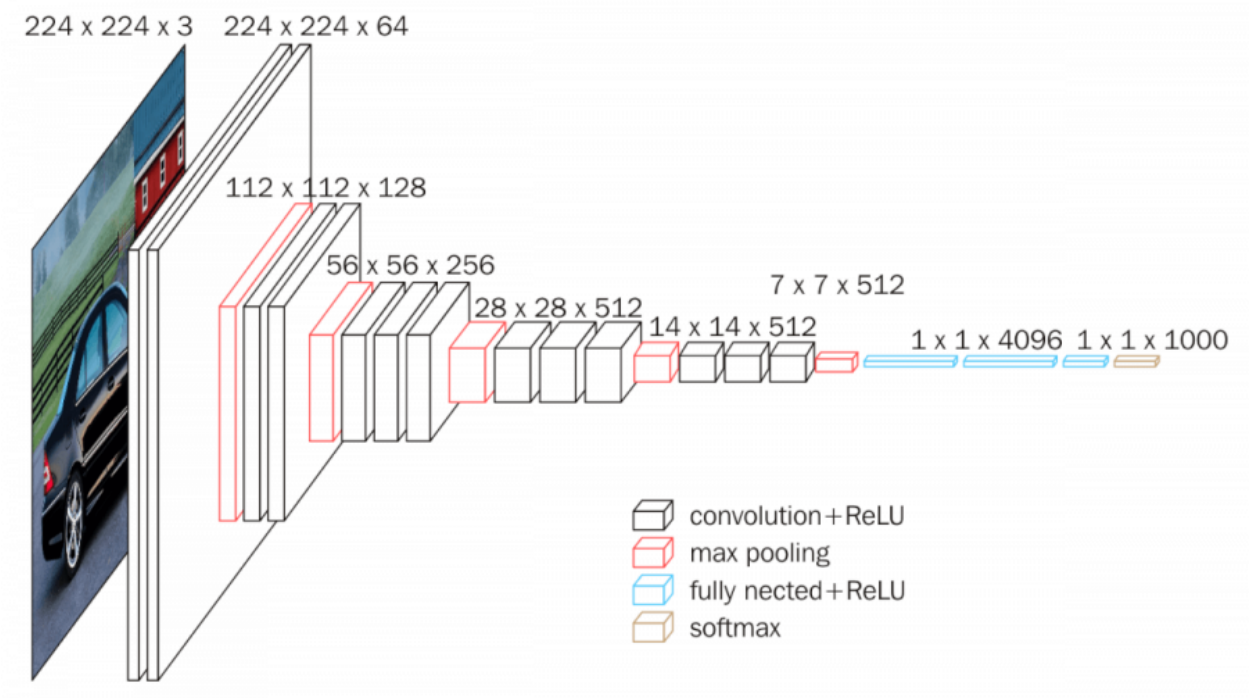
- Aim towards live emojification of facial expressions
- Dataset Definition: Working on Facial emotion recognition dataset FER-13.
- Developing the CNN based model that will be able to learn and extract features from our training dataset and identify the corresponding emotions.
- VIdeo capturing and frame extraction done using cv2 library.
- The model after training was tested on pre-recorded videos to check its performance.
- The model was able to annotate the faces with the label of their emotions based on their facial expressions.
- The accuracy and loss plots for training data were analyzed.

RAVDESS dataset

- gender balanced dataset
- 24 actors
- calm, happy, sad, angry, fearful, surprise emotions used
- Face only used for our CNN model
- Pre-processing using open-CV
- 2 fps used to extract frames

Model improvement

- Trained a new CNN model having similar structure as VGG models with 8 trainable layers
- VGG-16 architecture

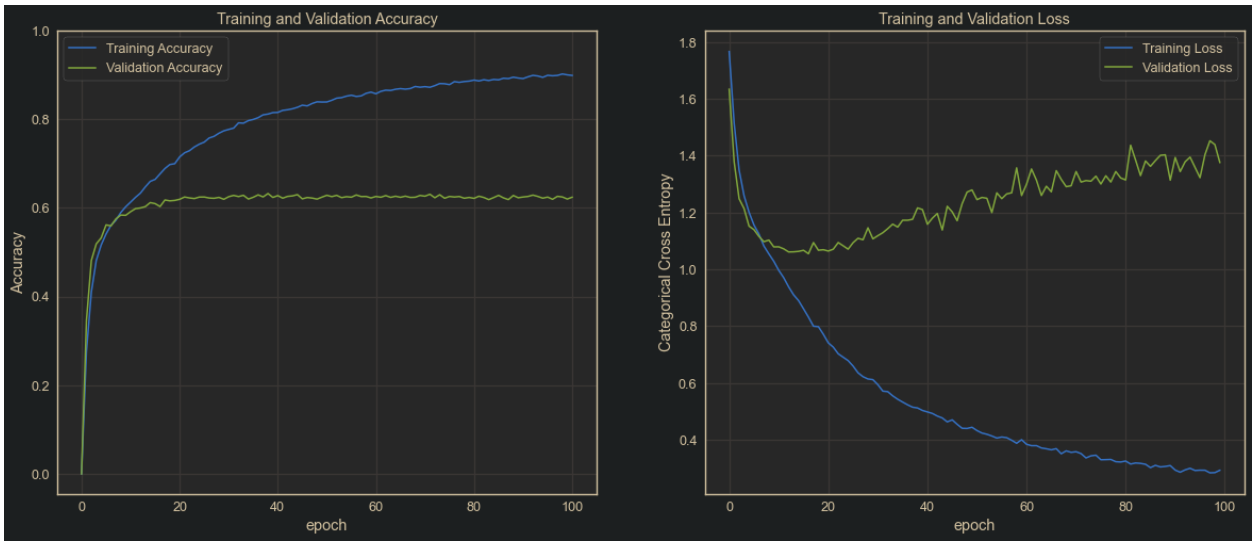


- New model architecture

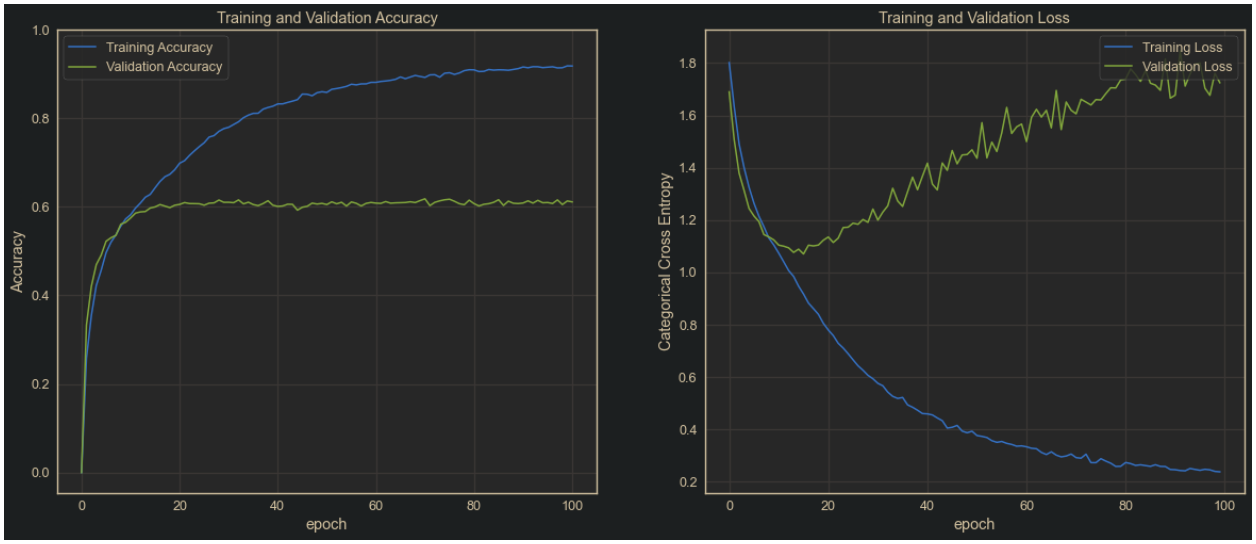
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 48, 48, 32)        320
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 48, 48, 32)        9248
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 24, 24, 32)        0
_________________________________________________________________
dropout (Dropout)            (None, 24, 24, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 24, 24, 64)        18496
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 24, 24, 64)        36928
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 12, 12, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 12, 12, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 12, 12, 128)       73856
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 12, 12, 128)       147584
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 6, 6, 128)         0
_________________________________________________________________
dropout_2 (Dropout)          (None, 6, 6, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 4608)              0
_________________________________________________________________
dense (Dense)                (None, 128)               589952
_________________________________________________________________
dropout_3 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 7)                 903
=================================================================
Total params: 877,287
Trainable params: 877,287
Non-trainable params: 0
```

- Trained and compared both models on FER and RAVDESS dataset

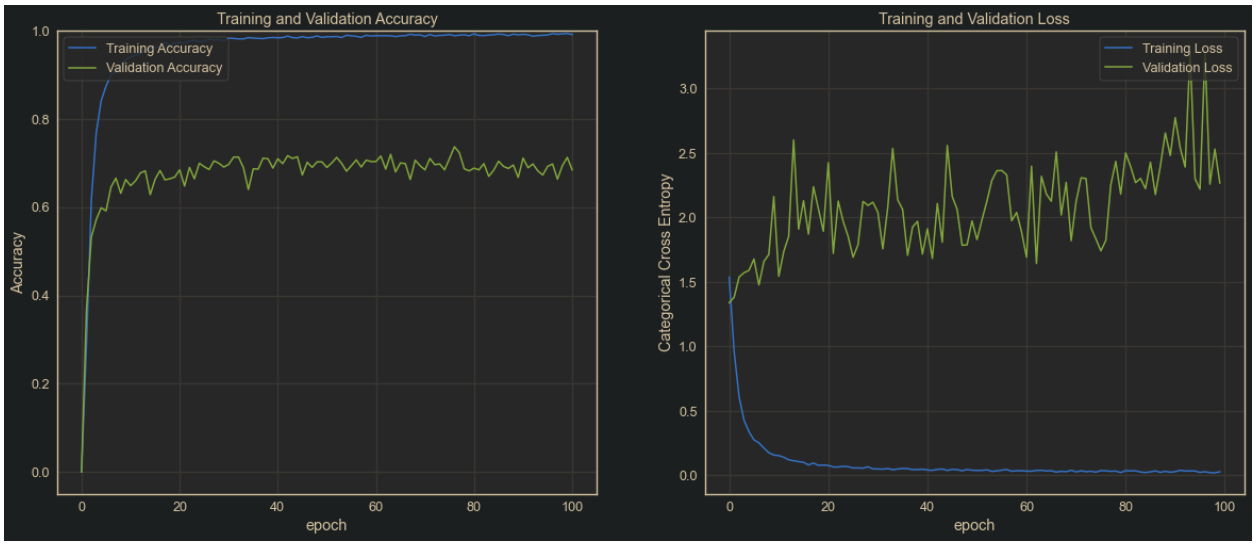
Old model on FER dataset



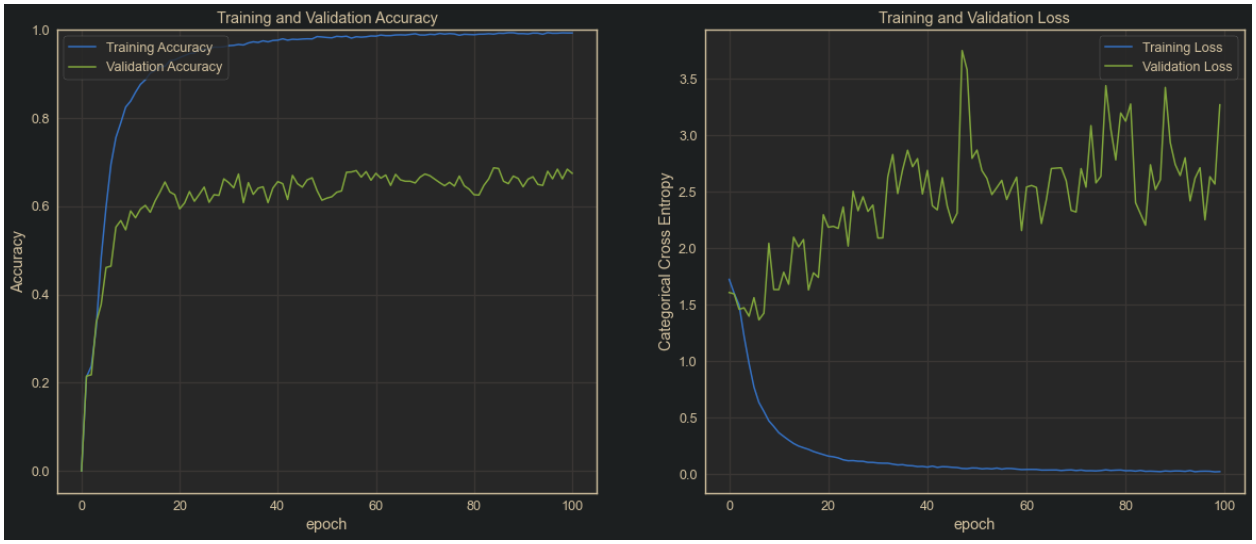
New model on FER dataset



Old model on RAVDESS dataset

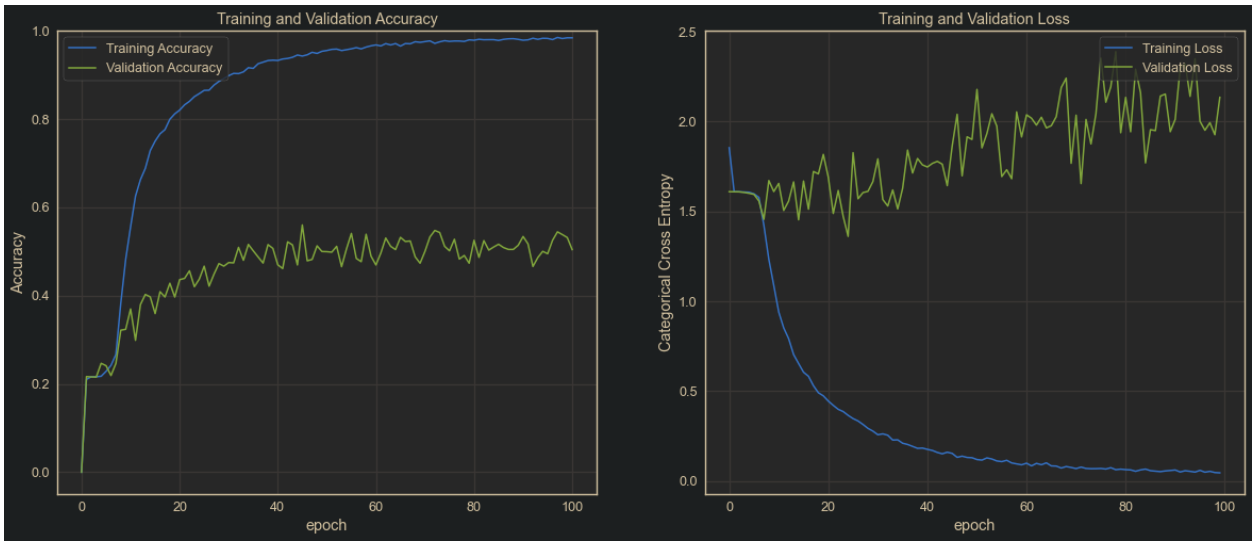


New model on RAVDESS dataset

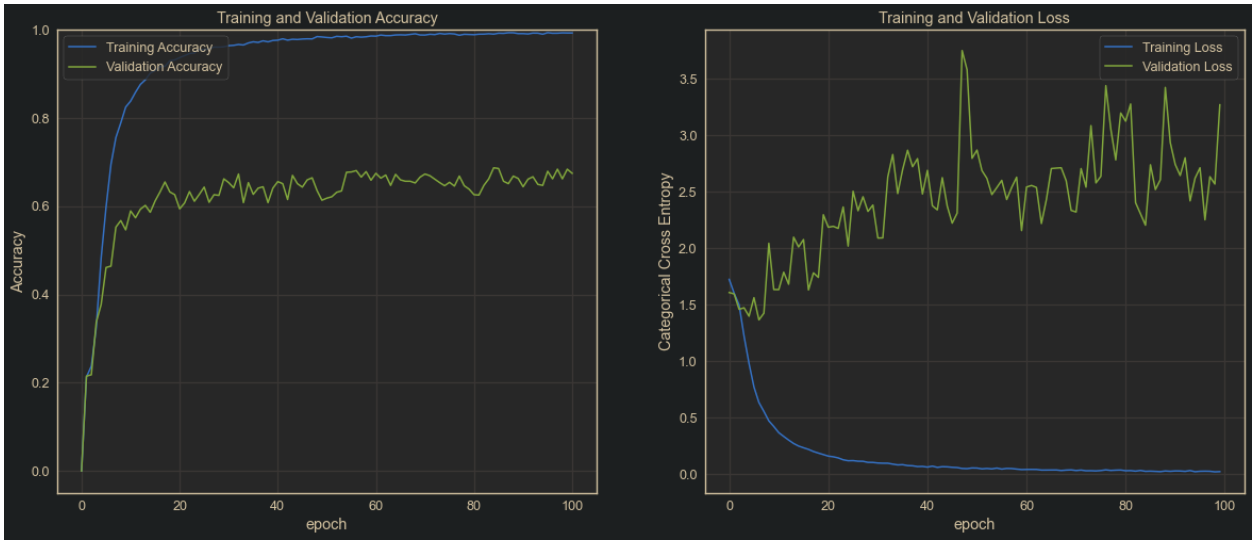


- New models accuracy didn't show great improvements - reason could be lack of training data

New model on RAVDESS dataset - 80-20 split



New model on RAVDESS dataset - 90-10 split



- As seen above, validation acc increases when number of training examples increase
- Number of parameters reduced from 2.3M to 0.8M - computationally faster

Emojification on live video

- Used OpenCV to capture live feed through webcam
- Model Predicts the emotion using extracted frame
- Used tkinter GUI toolkit to create a gui interface for showing the emoji along with the captured video/webcam.

Conclusion

- We learned how to implement the convolution neural networks for image classification.
- Used two different datasets and tried tuning the hyperparameters to get good accuracy.
- Learned how to use the opencv library for making it a real time task and applied the methods from tkinter toolkit to get a basic GUI interface.
- Model performs decently but could be better if we had a larger dataset.
- Datasets are not uniformly distributed throughout all categories and hence either categorical weight assignment method must be used or equal samples in all categories must be preferred.