Capstone Project - 5

Face Emotion Recognition

Vikaskumar Sharma

Content :

The Dilemma :

The Indian education landscape has been undergoing rapid changes for the past 10 years owing to the advancement of web-based learning services, specifically, eLearning platforms.

Global E-learning is estimated to witness an 8X over the next 5 years to reach USD 2B in 2021. India is expected to grow with a CAGR of 44% crossing the 10M users mark in 2021. Although the market is growing on a rapid scale, there are major challenges associated with digital learning when compared with brick and mortar classrooms. One of many challenges is how to ensure quality learning for students. Digital platforms might overpower physical classrooms in terms of content quality but when it comes to understanding whether students are able to grasp the content in a live class scenario is yet an open-end challenge.

The Dilemma contd…

In a physical classroom during a lecturing teacher can see the faces and assess the emotion of the class and tune their lecture accordingly, whether he is going fast or slow. He can identify students who need special attention. It provides data in the form of video, audio, and texts which can be analyzed using deep learning algorithms. Deep learning backed system not only solves the surveillance issue, but it also removes the human bias from the system, and all information is no longer in the teacher's brain rather translated in numbers that can be analyzed and tracked.

We will solve the above-mentioned challenge by applying deep learning algorithms to live video data. The solution to this problem is by recognizing facial emotions. The model should be able to real-time identify the emotions of students in a live class.

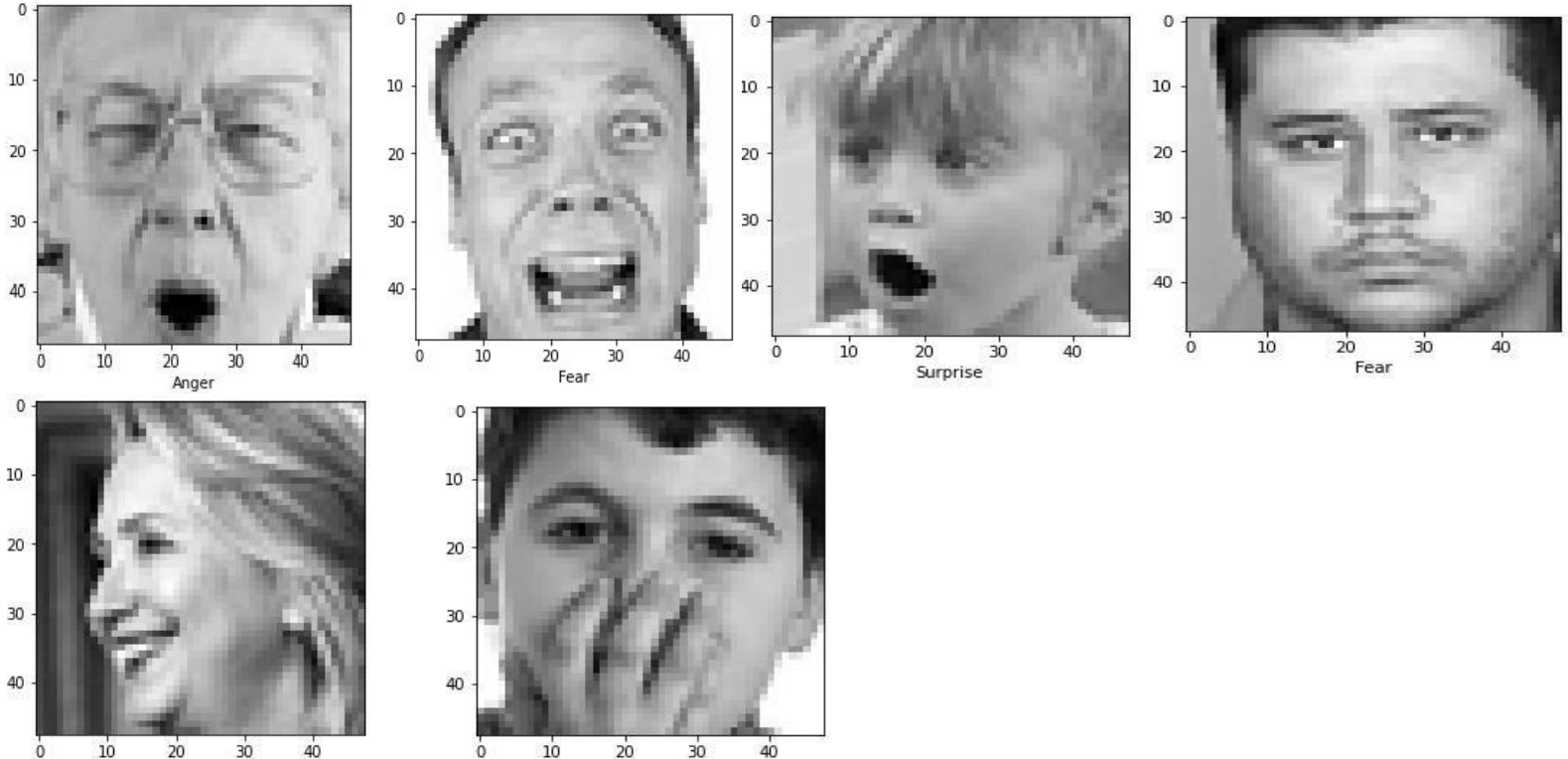
Dataset (FER)

This model is capable of recognizing seven basic emotions as following:

1. **Happy**
2. **Sad**
3. **Angry**
4. **Surprise**
5. **Fear**
6. **Neutral**
7. **Disgust**

EDA

Plotted images with their corresponding emotions :



Preparing dataset for modeling

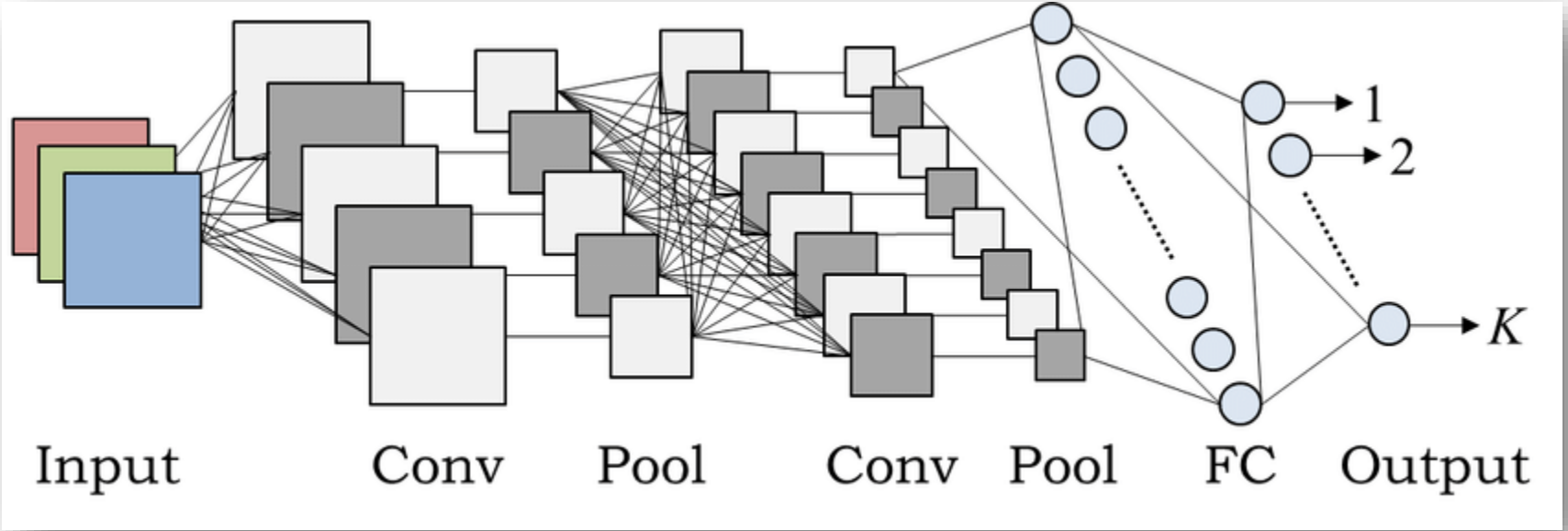
FER dataset have 7 different
Emotions and images in pixels
Form, converted pixels into images
And the dataset contains
28,821 images for training and
7,066 images for test purpose
which makes to 80 : 20 ratio.

Dependencies

1. Python 3
2. Tensorflow 2.0
3. Stream-lit
4. Streamlit-Webrtc
5. OpenCV
6. Numpy
7. Matplotlib
8. Pandas

Using CNN model

- A convolutional layer is **the main building block of a CNN**. It contains a set of filters (or kernels), parameters of which are to be learned throughout the training.
- Pooling layers are **used to reduce the dimensions of the feature maps.**
- **Fully Connected layers** in a neural networks are those layers where all the inputs from one layer are connected to every **activation unit** of the next layer.



Building the CNN model :

```
from tensorflow.keras.optimizers import Adam,SGD,RMSprop


no_of_classes = 7

model = Sequential()

#1st CNN layer
model.add(Conv2D(64,(3,3),padding = 'same',input_shape = (48,48,1)))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout(0.25))

#2nd CNN layer
model.add(Conv2D(128,(5,5),padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout (0.25))

#3rd CNN layer
model.add(Conv2D(512,(3,3),padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout (0.25))
```

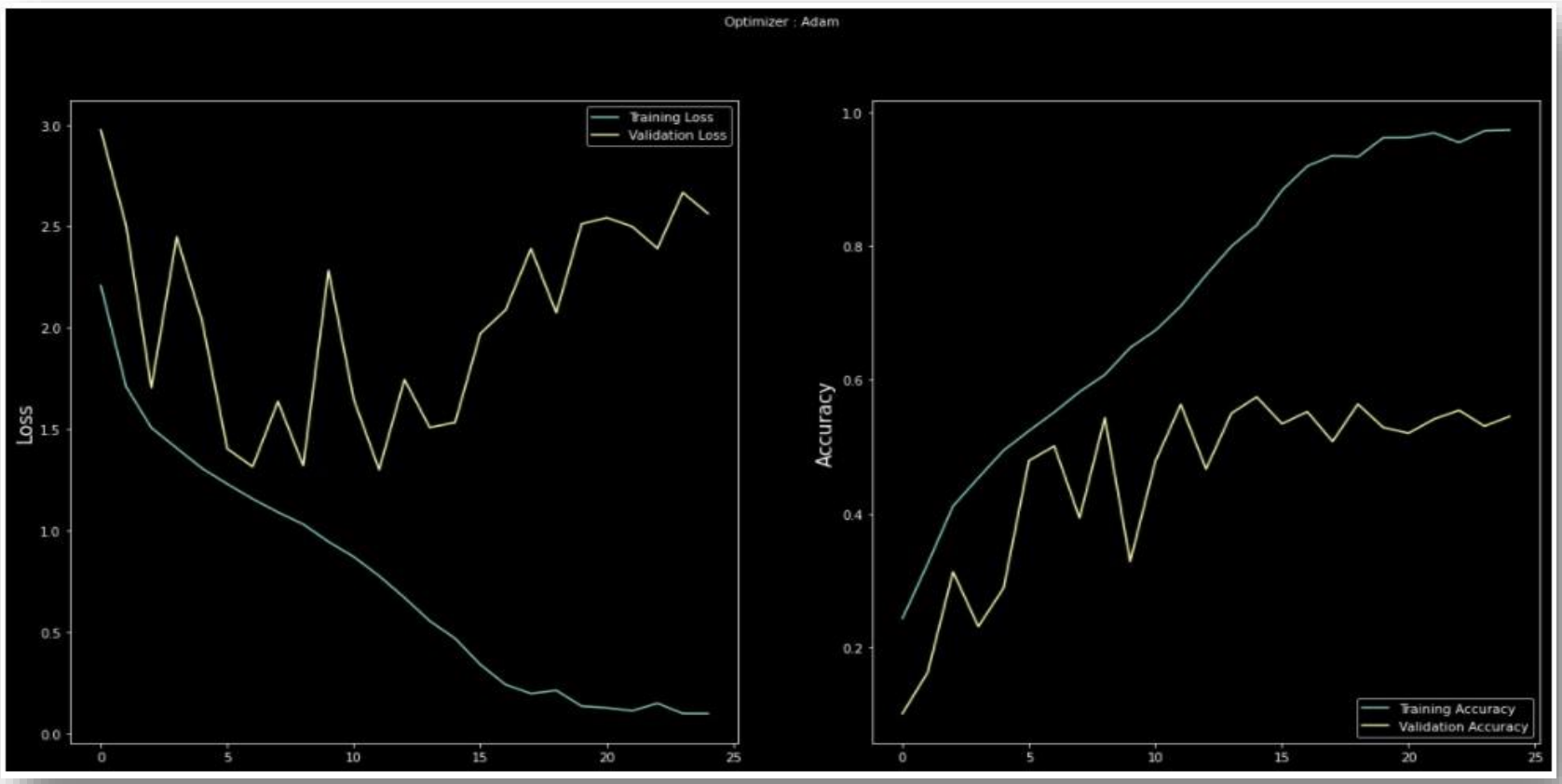
```
#4th CNN layer
model.add(Conv2D(512,(3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())

#Fully connected 1st layer
model.add(Dense(256))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))


# Fully connected layer 2nd layer
model.add(Dense(512))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))

model.add(Dense(no_of_classes, activation='softmax'))



opt = Adam(lr = 0.0001)
model.compile(optimizer=opt,loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()
```

Fitting the model with Training and Validation

In this model
we are using Adam optimizer.
The epochs used in the model are 48
batch size of 128
shuffling =True,
Validation split =0.2

```
Epoch 9: ReduceLROnPlateau reducing learning rate to 0.00020000000949949026.
225/225 [==============================] - 1416s 6s/step - loss: 0.9437 - accuracy: 0.6432 - val_loss: 1.1571 - val_accuracy:
0.5646 - lr: 0.0010
Epoch 9: early stopping
```

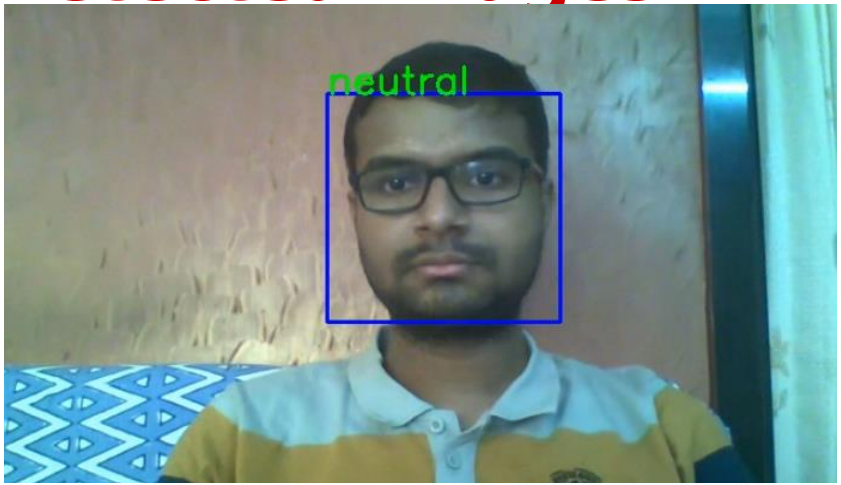
EDA(plotting loss and accuracy)



Testing the model contd…

1. As soon as you run the code a new window will pop up and your webcam will turn on.
2. It will then detect the face of the person, draw a bounding box over the detected person, and then convert the RGB image into grayscale & classify it in real-time
3. Deployed it on Heroku and stream-lit platform using Streamlit-Webrtc for the front end application

Detected images :



Challenges:

1. Large image dataset to handle
2. Couldn’t able to connect GPU with Jupyter notebook
3. Tired of using different models, finally found the best one
4. Continuous Runtime and Ram Crash due to large dataset
5. Deploying project at Heroku platform (tensorflow version is not supporting most of times but successfully deployed it)
6. Carefully tuned hyperparameters

Conclusion:

- I conclude this project by hoping that you got a fair idea and understood the whole pipeline on how you can make an emotion detection model
- We trained our model using Convolutional Neural Network (CNN) we just added layers with a channels and padding requirement in a sequential model just by calling add method.
- we also used Computer Vision as part of this model. Haarcascade is the package used from OpenCV to detect objects in other images.
- We trained the model with several images and then used the test images to see how the results match up. and we have taken epochs as 48 and we got the optimum score at 9th epoch
- For this model, the accuracy that we achieved for the validation set is 57%. To further increase the accuracy of the model, we can either expand the training dataset we have or increase the batch size for the model. Through these parameters, we can increase the model accuracy for this model.
- Model is identifying students emotions using minimum reference images and Successfully deployed web app of real-time webcam video feed on streamlit platform

THANK YOU