Capstone Project Submission

Instructions:
i) Please fill in all the required information.
ii) Avoid grammatical errors.

Team Member's Name, Email and Contribution:
Sampanna Mishra - mishrasampanna1998@gmail.com • Data Summary • Dependencies • Model Building • Model Evaluation • Web App and Cloud Deployment • Conclusion
Please paste the GitHub Repo link.
Github Link:- https://github.com/SampannaMishra/Face_Emotion_Recognition_Live_Stream Deployed Web App Link:- https://xcv-app.herokuapp.com/
CONTINUED

Please write a short summary of your Capstone project and its components. Describe the problem statement, your approaches and your conclusions. (200-400 words)

Emotion is a mental state associated with nervous system associated with feeling , perception,behavioral reactions and a degree of gratification or displeasure.Facial expressions are a form of nonverbal communication.One of the current application of artificial intelligence using neural networks is the recognition of faces in images and video for various applications.Face detection technology can come in handy in various practical life examples. Digital classrooms are conducted via video telephony software programs where it's not possible for educators to see all students and assess the mood. Because of this drawback, students are not focusing on content due to lack of surveillance.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

Data Summary:

The model is trained on the FER-2013 dataset which was downloaded from Kaggle.This dataset consists of 48x48 pixel grayscale images of faces. The training set consists of 28,709 images and the test set consists of 7,178 images. The dataset consists of 7 different facial expressions as follows: anger, disgust, fear, happy,surprise ,sad and neutral.

Dependencies:

- Python
- Tensorflow
- Keras
- Opencv
- Pre Trained Haar Cascade
- Streamlit
- Streamlit - webrtc

Model Building:

A Convolutional Neural Network, also known as CNN or ConvNet, is a class of neural network that specializes in processing data that has a grid-like topology, such as an image.

- The Convolution Neural Network (CNN) model has been used for processing the training images. A Customized CNN model has been built using Conv2D, Max Pooling, Batch Normalization, Dropout and Dense parameters.
- Activation function used is Relu and softmax.
- Adam optimizer has been used in the model.Initial number of epochs set for model training was 45.However , by using model callbacks the training stopped epoc 15 once the accuracy began to remain stagnant.
- The total parameters generated during the training process is 4,478,727.
- The output has 7 nodes because our output emotions are seven in number.

Model Evaluation:

A convolutional neural network can be evaluated using the 'evaluate' method. This method takes the test data as its parameters. Before this, the data is plotted on the console using 'matplotlib' library and 'imshow' methods.The accuracy versus epoch data is visualized.This is done using matplotlib library.The model is evaluated, and the loss and accuracy are determined.

Web App and Cloud Deployment

Streamlit which is an open source web framework has been used to build face emotion recognition web app.
OpenCV, an open source computer vision and machine learning software library has also been used for real time face reading.The model weights were saved in json and h5 file format which were later used.
Created a function FaceEmotion to detect multiple faces in video camera which further provides a bounding box around faces and predicts face

emotion.

Streamlit library does not provide the live capture feature itself ,instead uses a third party API.Therefore,I used streamlit-webrtc which helped to deal with real-time video streams. Then this model was deployed on the heroku platform.

Conclusion

Finally built a Face Emotion Recognition webapp using streamlit and deployed on heroku cloud, which predicts the face emotions on live webcam.The model created with CNN layers gave training accuracy 73.69% and validation accuracy 59.01% .