Live Class Monitoring System- Face Emotion Recognition

Sharad Tawade, Sagar Malik
Vinay Kumar
Data science trainee,
AlmaBetter

Abstract:

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.

In a physical classroom during a lecture, the 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. Digital classrooms are conducted via a video telephony software program (ex-Zoom) where it's not possible to see all students and access the mood. Because of this drawback, students are not focusing on content due to a lack of surveillance. Digital platforms have limitations in terms of physical surveillance but it comes with the power of data and machines which can work for you. Its data can be analyzed using deep learning algorithms which not only solves the surveillance issue but also removes the human bias from the system.

Keywords:eda, machine learning, mobile price range, classification

1.Problem Statement

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.

This is a few shots learning live face emotion detection systems. The model should be able to real-time identify the emotions of students in a live class.

2.Data Summary

The data comes from the past Kaggle competition "Challenges in Representation Learning: Facial Expression Recognition Challenge": we have defined the image size to 48 so each image will be reduced to a size of 48x48. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. Each image corresponds to a facial expression in one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). The dataset contains approximately 36K images.

3. Introduction

Facial emotion recognition is the process of detecting human emotions from facial expressions. The human brain recognizes emotions automatically, and software has now been developed that can recognize emotions as well. This technology is becoming more accurate all the time, and will eventually be able to read emotions as well as our brains do.

AI can detect emotions by learning what each facial expression means and applying that knowledge to the new information presented to it. Emotional artificial intelligence, or emotion AI, is a technology that is capable of reading, imitating, interpreting, and responding to human facial expressions and emotions.

4. Steps Involved

I. Exploratory Data Analysis:

The dataset contains approximately 36K images. Each image corresponds to a facial expression in one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). All the images are 48x48 pixels.

II. Data Image Generator:

Generate batches of tensor image data with real-time data augmentation.

III. Building Models:

We started our project with the concept of transfer learning i.e. using the pre-trained weights of ResNet50 for the training of the model and then fine-tuning them to increase the accuracy. Due to our dissatisfaction with this result, we switched to the Custom CNN model.

IV. Training The Model:

We trained the model with 40 epochs and callback as early stopping and ReduceLROnPlateau to avoid overfitting and reach the global minima.

V. Model Evaluation:

We evaluated the model using an accuracy plot and categorical cross-entropy loss and confusion matrix to find out in which category the model has inadequate performance and among which category the model is getting confused

VI. Model Deployment:

We have created a front-end using Streamlit for web apps and used streamlit-webrtc which helped to deal with real-time video streams. Image captured from the webcam is sent to the VideoTransformer function to detect the emotion. Then this model was deployed on the Heroku platform with the help of buildpack-apt which is necessary to deploy the OpenCV model on Heroku.

5. Models

Basic CNN architecture details:

● **Input layer** - The input layer in CNN should contain image data

● **Convo layer** - The convo layer is sometimes called the feature extractor layer because features of the image are get extracted within this layer

● **Pooling layer** - Pooling is used to reduce the dimensionality of each feature while retaining the most important information. It is used between two convolution layer

● **Fully CL** - Fully connected layer involves weights, biases, and neurons. It connects neurons in one layer to neurons in another layer. It is used to classify images between different categories by training and placed before the output layer

- **Output Layer** - The output layer contains the label which is in the form of a one-hot encoded.

I. DeepFace:

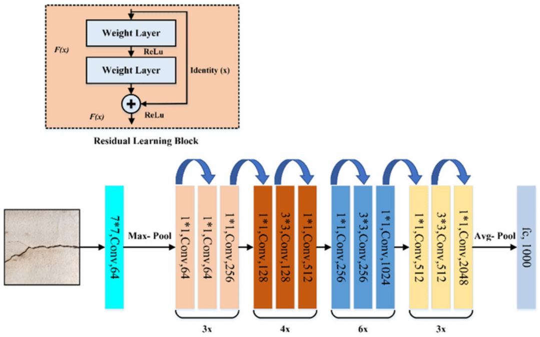
DeepFace is the most lightweight face recognition and facial attribute analysis library for Python. The open-sourced DeepFace library includes all leading-edge AI models for face recognition and automatically handles all procedures for facial recognition in the background.

The program employs a nine-layer neural network with over 120 million connection weights and was trained on four million images uploaded by Facebook users.

Deepface is used for face verification, face recognition, facial attribute analysis and real time face analysis.

II. Transfer Learning(ResNet50):

ResNet-50 is a convolutional neural network that is 50 layers deep. You can load a pre-trained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.



ResNet50 is a variant of the ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. It has 3.8 x 10^9 Floating points operations.

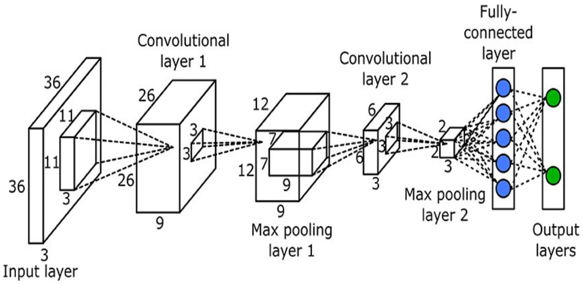
The ResNets were initially applied to the image recognition task but as mentioned in the paper that the framework can also be used for non-computer vision tasks to achieve better accuracy.

We started with ResNet50 and added 2 FC layers and trained the model by freezing all Conv layers except the last 4 and on the second run, we fine tuned the model by unfreezing all the layers. We got training accuracy of 43.69% and validation accuracy of 40.9% .

III. Convolutional Neural Network:

CNN's are powerful image processing, artificial intelligence (AI) that use deep learning to perform both generative and descriptive tasks, often using machine vision that includes image and video recognition and processing that is specifically designed to process pixel data.

A convolution network generally consists of alternate convolution and max-pooling operations. The output obtained after applying convolution operation is shrunk using max-pooling operation which is then used as an input for the next layer.



We define our CNN with the following global architecture:

- 4 convolutional layers
- 2 fully connected layers

Also, we use some common techniques for each layer

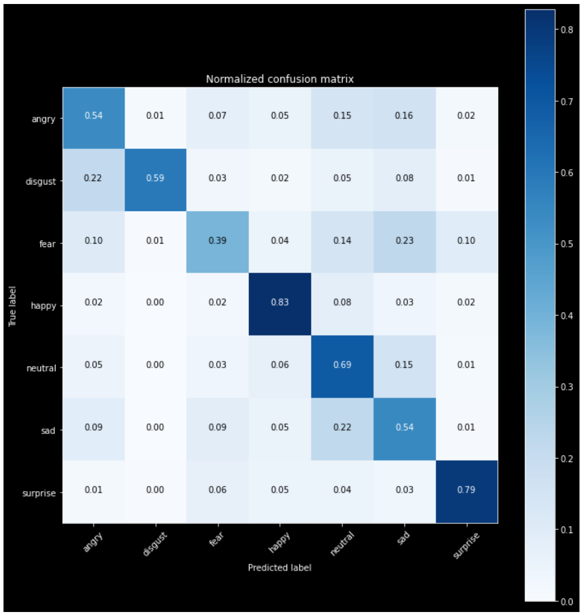
Batch normalization: improves the performance and stability of NNs by providing inputs with zero mean and unit variance. Dropout: reduces overfitting by randomly not updating the weights of some nodes. This helps prevent the NN from relying on one node in the layer too much.

6. Model Performance

I. Confusion Matrix:

The confusion matrix is a table that summarizes how successful the classification modelis at predicting examples belonging to various classes. One axis of the confusion matrix is the label that the model predicted, and the other axis is the actual label.

Our model is very good for predicting happy and surprised faces. However, it predicts quite poorly feared faces maybe because it confuses them with sad faces.it also gets confused between angry and disgusted faces.



II. Precision/Recall:

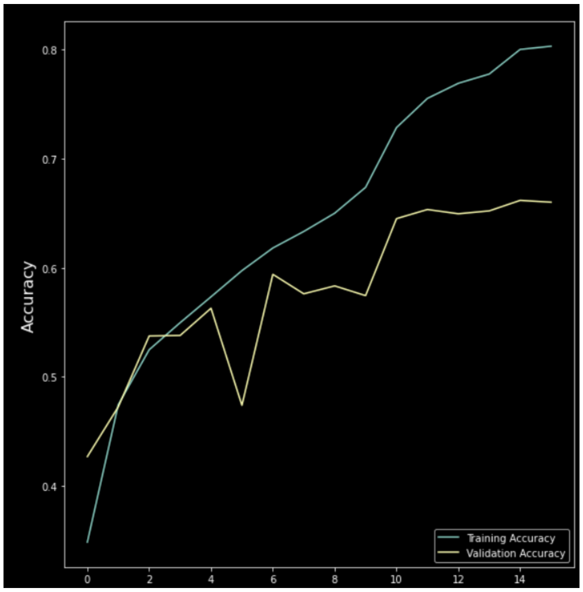
Precision is the ratio of correct positive predictions to the overall number of positive predictions : TP/TP+FP

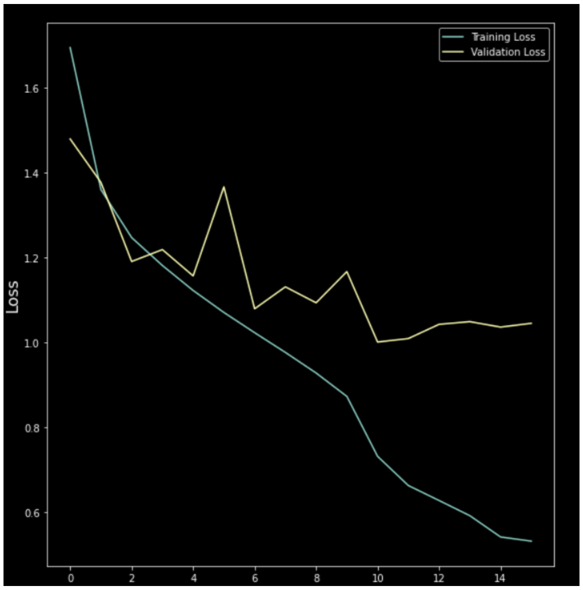
Recall is the ratio of correct positive predictions to the overall number of positive examples in the set: TP/FN+TP.

III. Accuracy:

Accuracy is given by the number of correctly classified examples divided by the total number of classified examples. In terms of the confusion matrix, it is given by: TP+TN/TP+TN+FP+FN

To evaluate the performance of your model. For our model, we have got 0.6454 which means our model is approx. 64.54% accurate.





IV. Loss (Categorical Cross Entropy):

The categorical cross-entropy loss function calculates the loss of an example by computing the following sum:

$$\text{Loss} = -\sum_{i=1}^{\substack{\text{output} \\ \text{size}}} y_i \cdot \log \hat{y}_i$$

Where y^i is the i-th scalar value in the model output,y_i is the corresponding target value, and the output size is the number of scalar values in the model output.

8. Conclusion:

Trained the neural network and we achieved the highest validation accuracy of 66%.

The Pre Trained Model didn't give appropriate results.

The application is able to detect face location and predict the right expression while checking it on a local webcam.

The front-end of the model was made using streamlit for webapp and running well on local webapp link.

Finally, we successfully deployed the Streamlit WebApp on Heroku and Streamlit share that runs on a web server.

Our Model can successfully detect face and predict emotion on live video feed as well as on an image.