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.

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. Digital classrooms are conducted via video telephony software program (ex- Zoom) where it’s not possible for medium scale class (25-50) to see all students and access the mood. Because of this drawback, students are not focusing on content due to lack of surveillance. While digital platforms have limitations in terms of physical surveillance but it comes with the power of data and machines which can work for you. It provides data in the form of video, audio, and texts which can be analysed 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 analysed and tracked.

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.

As the first step I divided the Project into 6 main steps:-

1. Data Collection & Understanding.
2. EDA.
3. Data Preprocessing.
4. Building CNN Model.
5. Testing the Model.
6. Conclusion.

In Data Collection & Understanding, I collected the data of face emotion recognition dataset from Kaggle by Jonathan Oheix.It contains images of 7 different emotions for training and testing purposes.The 7 different emotions are Happy,fear,disgust,sad,surprise,angry,neutral.For training 28821 images were there and for testing 7066 images were there which makes it to total 35887 images of different emotions in different files.

In EDA, we looked at different images of emotions.We find that for happy emotion we had around 8989 images which is the highest for any emotions ,followed by neutral 6198, for sad 6077, for fear 5121, for angry 4953, for surprise 4022, for disgust 547 images only.

In Data preprocessing,we load the image and then convert them to array form with the picture size of 48 x 48.It is the standard pixel size of the image.We use the imagedatagenerator library which Generate batches of tensor image data with real-time data augmentation.

Next we build the Convolutional neural network (CNN) model.It consist of Six layers which comprises of four CNN layer and two fully connected layers. The network consists of two convolutional layers with a filter size of 64 each. This is then followed by a max pooling layer. A dropout of rate 0.25 is applied to reduce overfitting. This is followed by a sequence of four convolutional layers which have a filter size of 256 each. A single max pooling layer follows these four layers with a dropout of rate 0.25. In order to convert the output into a single dimensional vector, the output of the previous layers was flattened. A fully connected layer with a L2 regularizer of penalty of 0.001 is then used alogn with an additional dropout of rate 0.5. Finally, a fully connected layer with a softmax activation function serves as the output layer.The kernel size, that is, the width and height of the 2D convolutional window is set to 3 x 3 and 5 x 5 for all convolutional layers. Each max pooling layer is two dimensional and uses a pool size of 2 x 2. This halves the size of the output after each pooling layer. All the layers bar the output layer used a ReLU activation function. The ReLU activation function is used here due to benefits such as sparsity and a reduced likelihood of vanishing gradient. The softmax activation function was used in the final output layer to receive the predicted probability of each emotion. This model provided a base accuracy of 0.57 on the testing set. The hyperparamters were then tuned, namely the batch size, the optimizer and the number of epochs. The model was set to run for 48 epochs. However, in the interest of saving time and computational power, the network was allowed to stop training if there was no change in the accuracy over consecutive epochs. That is, the network would stop training if there was no change in the accuracy over 4 continuous epochs. This saved both time and computational power, especially in cases where there was no change in the accuracy within the earlier epochs themselves. The decision turned out to be a good one as none of the models exceeded 9 epochs.

In Testing,The dataset was initially split into an 80%-training set and a 20%-testing set. During the testing phase, each of the trained networks was loaded and fed the entire testing set one image at a time. This image was a new one which the model had never seen before. The image fed to the model was preprocessed in the same way as detailed . Thus the model did not know already what the correct output was and had to accurately predict it based on its own training. It attempted to classify the emotion shown on the image simply based on what it had already learned along with the characteristics of the image itself. Thus in the end, it gave a list of classified emotion probabilities for each image. The highest probability emotion for each image was then compared with the actual emotions associated with the images to count the number of accurate predictions.

Finally, I concluded with following points:-

* **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 Heroku and streamlit platform**

Contributor Role:-

Vikaskumar Sharma.

1. **Defining problem statement**
2. **Preparing dataset for modelling**
3. **Exploratory Data Analysis (EDA)**
4. **Dependencies**
5. **Creating model**
6. **Fitting the model with training data and validation data**
7. **Testing the model**
8. **Detected images**
9. **Conclusion**

Github Link:- <https://github.com/VikasSharma3368/Netflix-Movies-TV-Shows-Clustering>

Drive Link:- <https://drive.google.com/drive/folders/1OlPfK2UyLej6qSNzZ0qd7j-E96NvOjLd?usp=sharing>