Live Class Monitoring System (Face Emotion Recognition)

Sayan Bandopadhyay
Data science trainee,
AlmaBetter, Bangalore

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. It provides data in the form of video, audio and texts which can be analyzed using the deep learning algorithms.

The motive here is to solve the above-mentioned challenge by applying deep learning algorithms to live video data. The experiment performed can help in understanding the overall student's sentimental behavior based on their face emotions. This can be used to analyze whether the students found the class interesting or not. This may further help to improve the web-based learnings.

Keywords: eLearning, deep learning, sentimental behavior, emotions.

Problem Statement

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 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.

Steps Involved

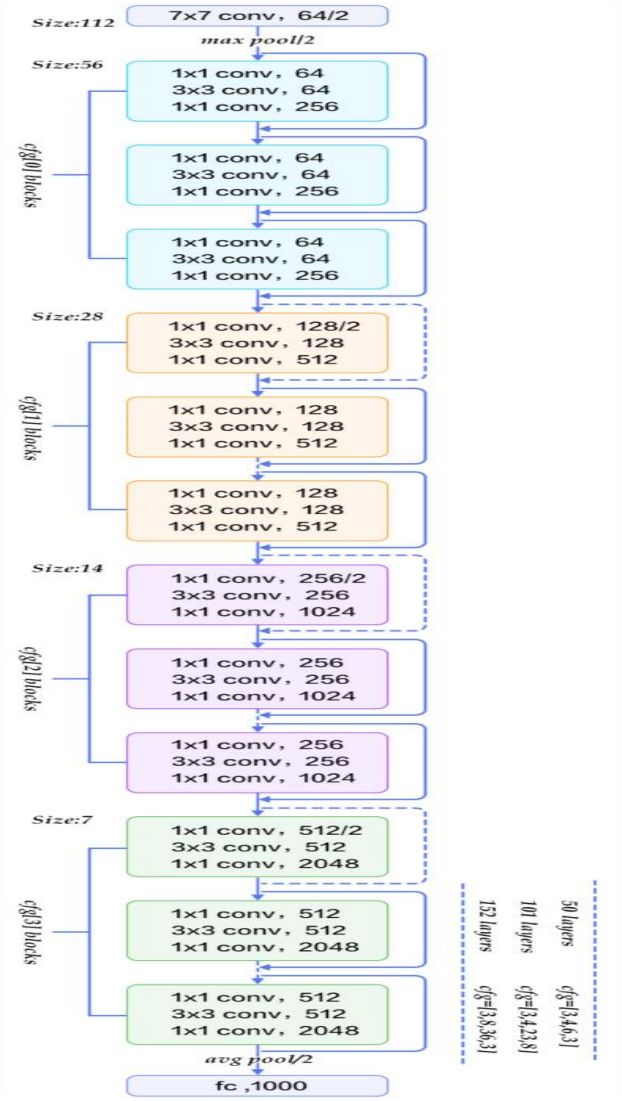
Following are the different steps been taken in the overall analysis process right from the beginning to the end:

1. **Framing Questions:** First step includes deep understanding of the problem statement and pin pointing the question arising based on the problem. This helps in setting the project objective and also helps in getting an idea of the process pathway that will lead to the conclusion.
2. **Accessing Dataset:** Second step is all about finding an appropriate source of required data and getting an access of it in order to carry out the project. In this case, the required dataset is downloaded from 'Kaggle'.
3. **Data Preparation:** Third step is basically about making the data usable, as in this case the dataset was in zip format so needed to perform some operations to unzip the data in the data directory.
4. **Data Visualization:** Fourth step is about data inspection and visualization which is a part of exploratory data analysis (EDA). Its all about checking what the dataset is actually comprised of and what it actually looks like. This step answers all the question regarding number of classes, total images in train and test dataset, number of images for each class etc.
5. **Data Pre-processing:** Fifth step is one of the most important step before model creation as its about tweaking the data so that it becomes a good fit for the model.
6. **Model Building:** It's the sixth step which has 3 sub parts: creating model layers, compiling the model layers and finally fitting the compiled model. It's fitted after taking the access of GPU in order to make the training process faster.
7. **Evaluating Model:** Seventh step comes after trying different models and coming up with best performing model out of all and evaluating it based on different evaluation metrics and confusion matrix.
8. **Saving Model:** This step includes saving the best performing model in order to use it in developing a web application.
9. **Creating and Deploying Web Application:** It's the last step which includes development of a web application which will be able to capture real time face and detect its emotion. Post creation of application, its deployed on a cloud server.

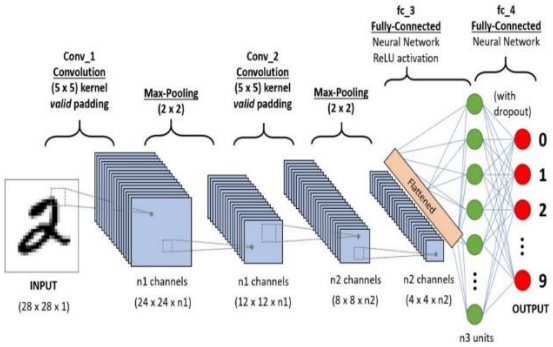
Algorithms

Post pre-processing, 3 models are built based on transfer learning and customized 'Convolutional Neural Network' model. Algorithms used are as follows:

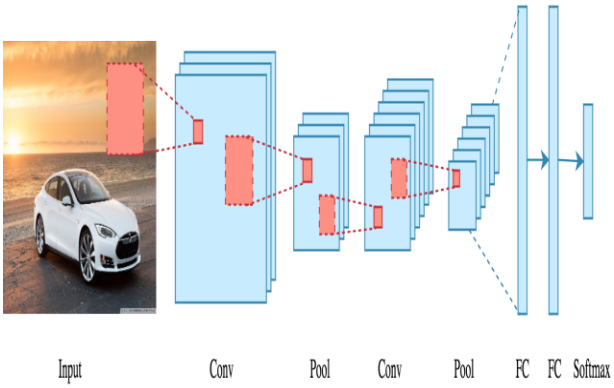
ResNet50 (Transfer Learning): First model is built using the transfer learning approach taking the use of a pre-trained architecture named as 'ResNet50'. ResNet50 is a variant of ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. It has 3.8 x 10^9 floating points operations. It is a widely used ResNet model. ResNet, short for Residual Networks is a classic neural network used as a backbone for many computer vision tasks. Following is the diagrammatic representation of ResNet50 architecture:



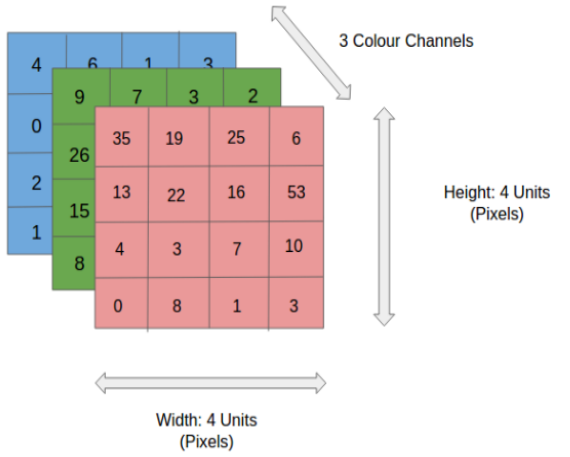
Convolution Neural Networks (CNN): One of the most popular deep neural networks is Convolutional Neural Networks.



In deep learning, CNN/ConvNet is a class of deep neural networks most commonly applied to analyze visual imagery. It uses a special technique called Convolution. In mathematics, convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.



An RGB image is nothing but a matrix of pixel values having three planes whereas a grayscale image is the same but it has a single plane.

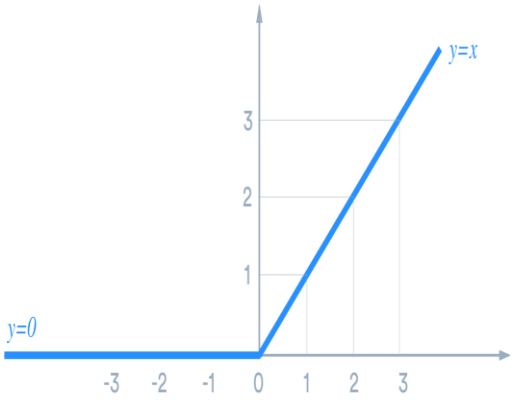


Convolutional neural networks are composed of multiple layers of artificial neurons. Artificial neurons, a rough imitation of their biological counterparts, are mathematical functions that calculate the weighted sum of multiple inputs and outputs an activation value. Each layer generates several activation functions that are passed on to the next layer. The first layer usually extracts basic features such as horizontal or diagonal edges. This output is passed on to the next layer which detects more complex features such as corners or combinational edges. Moving deeper into the network it can identify even more complex features such as objects, faces, etc.

Parameters and Hyper-Parameters

- **Activation Function**: In artificial neural networks, the activation function of a node defines the output of that node given an input or set of inputs. Two activation function is used in the model, 'Rectified Linear Unit (ReLu)' and 'SoftMax'.

1. ReLu: The rectified linear activation function or ReLU is a non-linear function or piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. Graphically, its represented as follows:



Mathematically its expressed as follows:

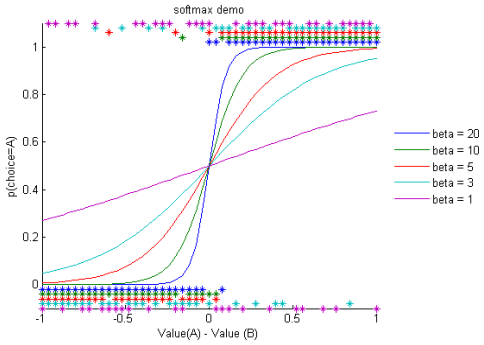
$$f(x) = max(0, x)$$

2. SoftMax: SoftMax is a mathematical function that converts a vector of numbers into a vector of probabilities, where the probabilities of each value are proportional to the relative scale of each value in the vector. Specifically, the network is configured to output N values, one for each class in the classification task, and the SoftMax function is used to normalize the outputs, converting them from weighted sum values into probabilities that sum to one. Each value in the output of the SoftMax function is interpreted as the probability of membership for

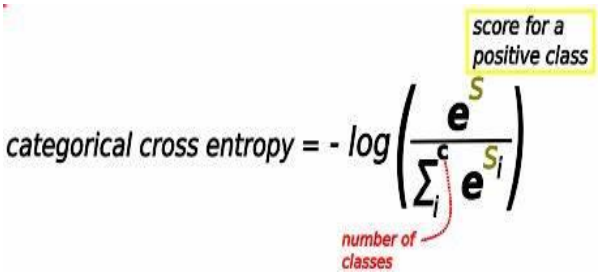
each class. Mathematically it is expressed as follows:

$$softmax(z_i) = \frac{exp(z_i)}{\sum_j exp(z_j)}$$

Graphically its represented as follows:



- **Loss Function**: The loss function is used to compute the quantity, that the model should seek to minimize during training. 'Cross entropy' loss function is an optimization function which is used in case of training a classification model which classifies the data by predicting the probability of whether the data belongs to one class or the other class. Specifically, 'categorical cross-entropy' is used in this case. Categorical cross-entropy is a loss function that is used for single label categorization. This is when only one category is applicable for each data point. In other words, an example can belong to one class only. The block before the Target block must use the activation function SoftMax.



- **Optimizer:** Adam optimizer is used in this case which is based on 'Adam optimization algorithm'. Adam is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data. The name 'Adam' is derived from adaptive moment estimation.

- **Learning Rate:** It's the rate of learning or speed at which the model learns and is controlled by the hyperparameter. It regulates the amount of allocated error with which the model's weights are updated each time they are updated, such as at the end of each batch of training instances. In this particular case, default Adam learning rate is used (0.001) for both ResNet50 and 1st CNN model but the learning rate used for 2nd CNN model is 0.0001 just to check if it manages to improve the accuracy.

- **Batch Size:** It refers to the number of training examples utilized in one iteration. Batch size used in this case is 32.

- **Epochs:** In terms of artificial neural networks, an epoch refers to one cycle through the full training dataset. In this

case, all the models are trained for 50 epochs.

Model Performance

In this case, model performance is judged based on the following:

- Accuracy
- Confusion matrix
- Area under ROC curve (AUC)

Accuracy

Accuracy is the given number of correctly classified examples divided by total number of classified examples. In terms of confusion matrix, it's given by,

(TP+TN) / (TP+TN+FP+FN)

Confusion Matrix

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

Area under ROC curve (AUC)

ROC curves use a combination of the true positive rate (the proportion of positive examples predicted correctly known as Recall) and false positive rate (the proportion of negative examples predicted incorrectly) to build up a summary picture of the classification performance.

Conclusion

Right from understanding the problem statement, extracting and preparing the dataset, visualizing the data contents, data pre-processing, building different models, evaluating model performance, saving the best performing model and finally creating a web application, the project came to an end with a satisfactory result.

References

- Analytics Vidhya
- GeeksforGeeks
- JavaTpoint
- Codecademy
- Daniel Llatas Spiers Research Paper