Face Emotion Recognition

By Vithika Karan
Data Science Trainee
AlmaBetter, Bangalore

:Abstract:

Due to the expansion of web-based learning services, notably eLearning platforms, the Indian education scene has been undergoing fast changes for the past ten years.
But, with all of the advantages of web-based learning, there are also some drawbacks. One of the numerous issues is ensuring that students receive high-quality education. Even in live teacher-monitored lectures, determining if a student can absorb the subject is extremely challenging.
Recognizing facial emotions is the solution to this dilemma.

Artificial intelligence's ability to bridge the gap between human and computer skills has substantially improved.
The field of computer vision is one of several such disciplines.
Facial Emotion Recognition (FER) is a computer vision application and technology that analyses facial expressions in both static photos and videos to disclose information about a person's emotional state. For a live webcam broadcast, the work here trains a model and predicts emotions.

Keywords:* *CNN, Deep Learning, Model Architecture, FER

1. Problem Definition and Methods:

1.1 Problem Statement

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 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 video telephony software program (exZoom) where it's not possible for medium scale classes (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. 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 systems 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.

1.2 Business Problem Analysis

E-learning is a learning approach that is based on formalized instruction but uses electronic resources. While education can take place in or out of the classroom, E-learning is primarily dependent on the use of computers and the Internet. E-learning is a network-enabled transfer of skills and knowledge in which education is delivered to a large number of people at the same time or at different periods. Previously, it was not widely recognised because it was considered that this method lacked the human factor necessary for learning.

However, with quick technological advancements and advancements in learning methods, it is now widely accepted.

Without a question, it is critical to advance the concept of non-electronic education through the use of books and lectures, but the value and effectiveness of technology-based learning cannot be overlooked. The human brain is thought to be capable of remembering and relating to what is seen and heard in moving pictures or films. Visuals, in addition to maintaining the student's attention, have been discovered to be kept by the brain for extended lengths of time. Agriculture, medicine, education, services, business, and government institutions are all adjusting to the concept of E-learning, which aids in a country's advancement.

Aside from all of these benefits, e-learning has a number of disadvantages. Some of them include turning boring subject matter into engaging e-learning experiences, staying current with latest technology, and designing e-learning courses for diverse generations, but the most crucial is a lack of learner engagement and motivation.

Regrettably, not every online learner will be completely dedicated to the e-learning experience. They could be preoccupied, distracted, or simply unmotivated. It's difficult for educators and instructors to tell whether or not a student understands the topics being taught, especially when the number of students participating is vast compared to physical classrooms where teachers can observe and interact with them.

Face emotion detection can be used in web-based learning platforms like Zoom as a solution to this problem. The emotions of students can be tracked and studied in a systematic way, reducing the human bias as well.

1.3 Algorithms and Methods

Deep learning techniques rely on neural networks, which are a subset of machine learning. They're made up of node levels, each of which has an input layer, one or more hidden layers, and an output layer.

For classification and computer vision tasks, convolutional neural networks (ConvNets or CNNs) are more commonly used.

1.3.1 Convolutional Neural Networks:

The higher performance of convolutional neural networks with picture, speech, or audio signal inputs sets them apart from conventional neural networks. They are divided into three sorts of layers:

- Convolutional layer

- Pooling layer
- Fully-connected (FC) layer

A convolutional network's first layer is the convolutional layer. While further convolutional layers or pooling layers can be added after convolutional layers, the fully-connected layer is the last layer. The CNN becomes more complicated with each layer, detecting larger areas of the image. Earlier layers concentrate on basic elements like colors and borders. As the visual data travels through the CNN layers, it begins to distinguish larger elements or features of the item, eventually identifying the target object.

Convolutional Layer

The convolutional layer is the most important component of a CNN because it is where the majority of the computation takes place. It requires input data, a filter, and a feature map, among other things. A feature detector, also known as a kernel or a filter, will traverse across the image's receptive fields, checking for the presence of the feature. Convolution is the term for this procedure.

The feature detector is a two-dimensional (2-D) weighted array that represents a portion of the image. The filter size, which can vary in size, is usually a 3x3 matrix, which also affects the size of the receptive field.

After that, the filter is applied to a portion of the image, and a dot product between the input pixels and the filter is calculated. After that, the dot product is loaded into an output array. The filter then shifts by a stride, and the procedure is repeated until the kernel has swept across the entire image. A feature map, activation map, or convolved feature is the ultimate output of a series of dot products from the input and the filter.

Pooling Layer

Pooling Layer reduces the number of parameters in the input by performing dimensionality reduction. The pooling process sweeps a filter across the entire input, similar to the convolutional layer, however this filter does not have any weights. Instead, the kernel uses an aggregation function to populate the output array from the values in the receptive field. Pooling can be divided into two categories:

Max pooling: The filter selects the pixel with the highest value to send to the output array as it advances across the input. In comparison to average pooling, this strategy is employed more frequently.

Average pooling: The filter calculates the average value inside the receptive field as it passes across the input and sends it to the output array.

While a lot of information is lost in the pooling layer, it also has a number of benefits to CNN. They help to reduce complexity, improve efficiency, and limit risk of overfitting.

Fully-Connected Layer

The fully-connected layer's name is self-explanatory. In partially linked layers, the pixel values of the input image are not directly connected to the output layer, as previously stated. Each node in the output layer, on the other hand, connects directly to a node in the previous layer in the fully-connected layer.

This layer performs classification tasks based on the features retrieved by the previous layers and their various filters. While convolutional and pooling layers typically utilize ReLu functions to categorize inputs, FC layers typically use a softmax activation function to provide a probability from 0 to 1.

Batch Normalization

Batch normalization is a network layer that allows each layer to learn more independently. It's used to make the output of the previous layers more normal. In normalization, the activations scale the input layer. Learning becomes more efficient when batch normalization is utilized, and it can also be used as a regularization to prevent model overfitting. To standardize the inputs or outputs, the layer is added to the sequential model. It can be utilized at numerous points within the model's layers. It is frequently put immediately following the definition of the sequential model and before the convolution and pooling layers.

Dropout Layer

A Dropout layer is another common feature of CNNs. The Dropout layer is a mask that nullifies some neurons' contributions to the following layer while leaving all others unchanged. A Dropout layer can be applied to the input vector, nullifying some of its properties; however, it can also be applied to a hidden layer, nullifying some hidden neurons.

Dropout layers are critical in CNN training because they prevent the training data from being overfit. If they aren't there, the first batch of training data has a disproportionately large impact on learning. As a result, learning of traits that occur only in later samples or batches would be prevented.

Let's say we show CNN ten photographs of a circle in a row during training. Because the CNN will not learn that straight lines exist, it will be perplexed if we subsequently show it an image of a square. We can avoid these situations by incorporating Dropout layers into the network's architecture to prevent overfitting.

1.3.2 Data Augmentation

Image augmentation is a method of altering original images by applying various transformations to them, resulting in many altered copies of the same image. However, depending on the augmentation techniques you use, such as shifting, rotating, and flipping, each copy is unique in some ways.

Applying these minor changes to the original photograph does not modify its target class; rather, it provides a fresh viewpoint on capturing the object in real life. As a result, we frequently employ it in the development of deep learning models.

These picture augmentation techniques not only increase the amount of your dataset, but they also provide a level of variance to it, allowing your model to generalize better on data that hasn't been seen before. When the model is trained on new, slightly altered photos, it becomes more robust.

The ImageDataGenerator class in Keras makes it simple to add data to your photos. It offers a variety of augmentation options, including standardization, rotation, shifts, flips, brightness changes, and more.

The key advantage of utilizing the Keras ImageDataGenerator class, however, is that it is intended to give real-time data augmentation. That is, it creates augmented images on the fly while the model is still being trained. At each epoch, the ImageDataGenerator class guarantees that the model receives new variations of the images.

However, it merely returns the altered photos and does not include them in the original corpus. If this were the case, the model would be exposed to the original images many times, causing our model to overfit.

2. Introduction to FER

Facial Emotion Recognition is a system that analyzes attitudes from a variety of sources, including photos and videos. It's part of the family of technologies, which researches computers' ability to recognise and interpret human emotions. Facial expressions are nonverbal means of communication that reveal human emotions. Decoding such emotional responses has been a research topic in psychology for decades. Face detection, facial expression detection, and expression classification to an emotional state are the three processes in the FER analysis.

Talking from a business's point of view, FER has applications in the following areas:

- Industries invest a lot of money on monitoring, consumer feedback, and surveys, but the results are frequently unsatisfactory. In order to improve their products and services, FER can assist in these areas without bothering their clients.
- Analyze customers' feelings when shopping, with a focus on either items or how they are displayed in the store, or restaurants.
- Analyzing crime scene footage for possible reasons in a crime
- Analyzing facial expressions to predict individual reactions to movies
- Identify depression in the elderly
- Observe the patient's condition during hospitalization
- Look for signs of exhaustion in the driver in cab services industries.

3. Methodology and Results:

3.1 Data Summary

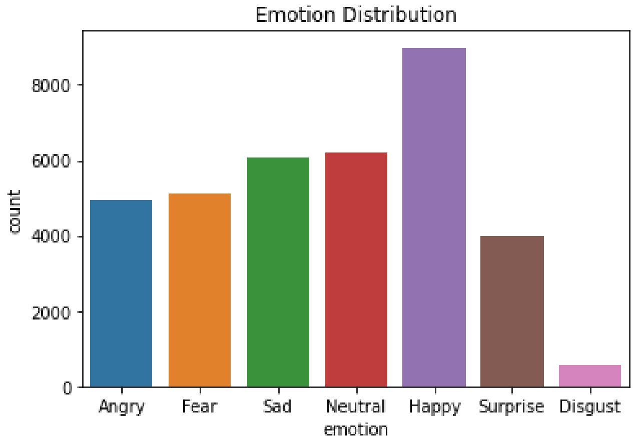
FER2013 (Facial Expression Recognition 2013) Dataset

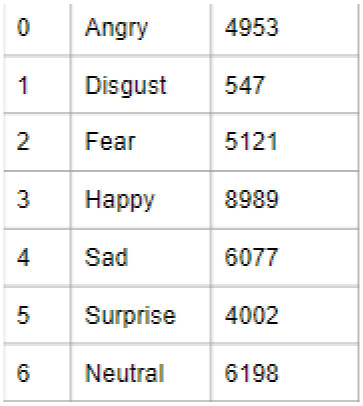
The data consists of 35887 grayscale images of faces at a resolution of 48x48 pixels. The faces have been automatically registered such that they are more or less centered in each image and take up around the same amount of area.

It has seven categories based on the emotion expressed in the facial expression (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). There are 28,709 examples in the training set and 3,589 examples in the public and private test sets.

3.2 Exploratory Data Analysis

The distribution of the emotion classes in the dataset is shown below.
It's worth noting that the number of cheerful photographs is the highest, while the number of disgusted images is the lowest.
Because there are so few data points in comparison to other classes, training the model for disgust will be difficult.





0	Angry	4953
1	Disgust	547
2	Fear	5121
3	Happy	8989
4	Sad	6077
5	Surprise	4002
6	Neutral	6198

3.3 Data Cleaning and Preprocessing

Data was clean and had no null values. Data preprocessing is done in order to transform the data into a form, understandable by the model. The preprocessing included splitting the dataset into training, validation and test sets. The pixels were given in the form of string objects, they were splitted and transformed into arrays,furthermore the data was prepared for the next step i.e. data augmentation. The method flow from directory for Image Data Augmentation expects that images belonging to different classes are present in different directories but are inside the same parent directory. The ImageDataGenerator class in Keras allows users to augment images while training the model. When utilizing the flow from directory() technique, the directory structure is crucial. The flow from directory() function assumes that the root directory has at least two folders, one for train and the other for test. The train folder should be included in subdirectories, each with photographs from different classes.
A single folder containing all test photos should be

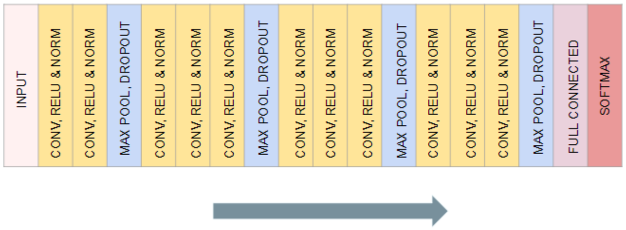
included in the test folder. Here, train and validation folders were created and the test data was kept aside to evaluate at the end.

3.4 Data Augmentation

Data Augmentation was carried out with Keras Image Data Generation, flow from directory method. The training data was rescaled, horizontally flipped, width changed, a little rotated. On the other hand, validation data were just rescaled and then the train and validation image data generators were fed to the CNN model.

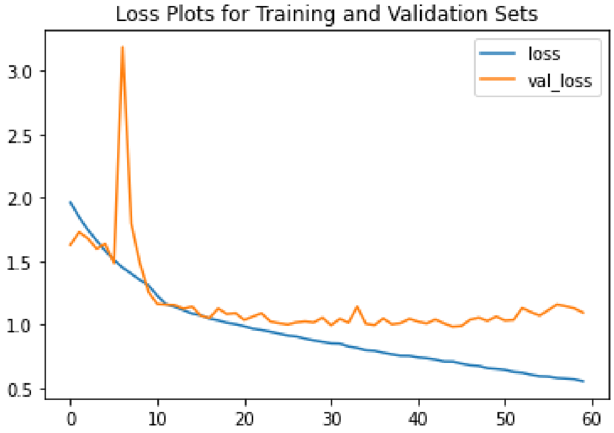
3.5 Model Building

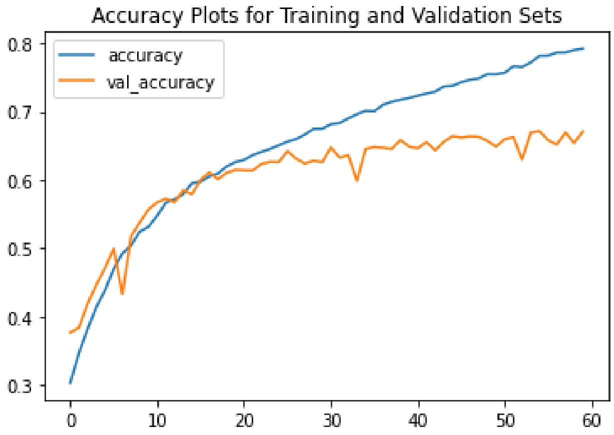
In Keras, the simplest way to build a model is sequential. It allows you to layer-by-layer construct a model. To add layers to our model, we use the 'add()' function.
The model architecture used in this project has four blocks. The first block contains two convolutional layers with “reLU” activation function followed by max pooling and dropout layer. The rest of the blocks contain three convolutional layers followed by max pooling and dropout layer. The kernel size used here is 3*3. Lastly, there’s a fully connected layer followed by a softmax activation function to classify the emotions. Below is a simple representation of the model architecture used here.



3.6 Model Performance and Evaluation

Here are the results and plots for loss and accuracy for both training and validation data.

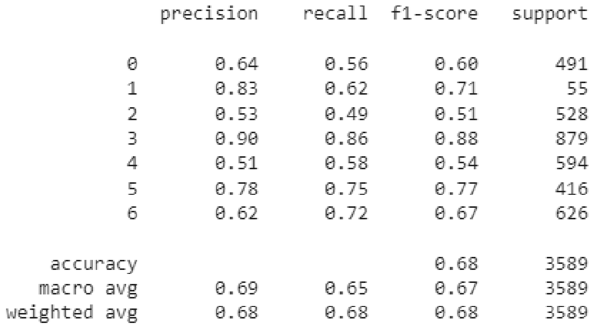




The best model weights were restored, which gave the following results for training and validation datasets.

loss: 0.7075 - accuracy: 0.7381 - val_loss: 0.9825 - val_accuracy: 0.6643

The best model weights gave a validation accuracy of 0.66 which is pretty great considering the unbalanced nature of the classes and less training data.

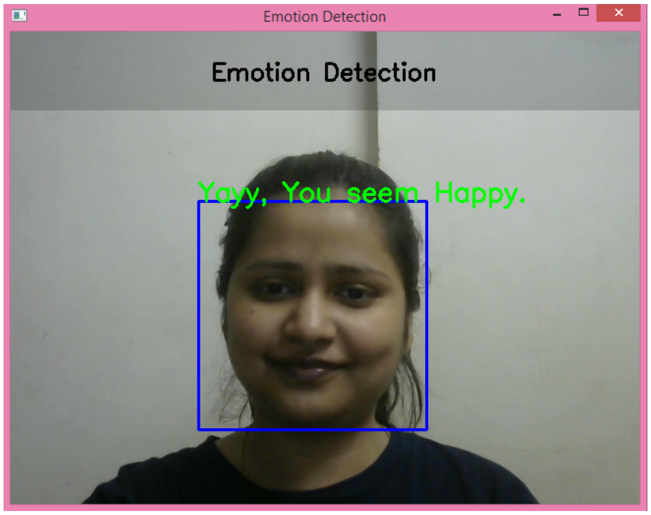


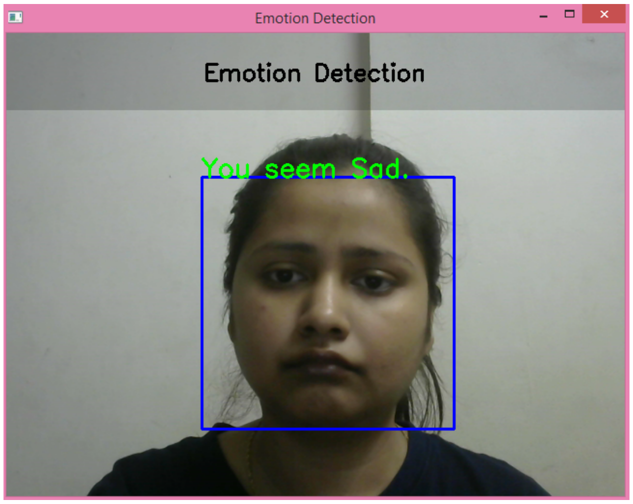
```
              precision    recall  f1-score   support

           0       0.64      0.56      0.60       491
           1       0.83      0.62      0.71        55
           2       0.53      0.49      0.51       528
           3       0.90      0.86      0.88       879
           4       0.51      0.58      0.54       594
           5       0.78      0.75      0.77       416
           6       0.62      0.72      0.67       626

    accuracy                           0.68      3589
   macro avg       0.69      0.65      0.67      3589
weighted avg       0.68      0.68      0.68      3589
```

Here are the results for model prediction on the test data which was kept aside. The model gave an accuracy of 0.68. This indicates the model is generalizing well on unseen data as well.

3.7 Real Time Emotion Detection

Real time emotion detection is an application of facial emotion recognition in which the prediction by the model happens on the live webcam feed. Using open source python libraries such as OpenCV-python the detector was created and here are few results from locally testing the application.





3.8 Streamlit Web Application

Streamlit is an open-source python framework for building web apps for Machine Learning and Data Science. We can instantly develop web apps and deploy them easily using Streamlit. Streamlit allows you to write an app the same way you write a python code. Streamlit makes it seamless to work on the interactive loop of coding and viewing results in the web app.

A web application for real time emotion detection was created using Streamlit to scale the use of the model.

3.9 Model Deployment

Deployment of machine learning models, or simply, putting models into production, means making your models available to other systems within the organization or the web, so that they can receive data and return their predictions.

In simpler terms, model deployment is the process of scaling the application to the world. So that anybody can access your work from anywhere with an internet connection.

The model was deployed on two platforms in this project. Heroku and AWS EC2 Instance.

3.9.1 Heroku

Challenges:

Heroku provides free services to deploy models on its cloud platform but with a limit; reducing the size of the slug generated by all the dependencies and model itself was a challenge. Since the slug size is larger than the soft limit provided by Heroku which is 300 MB, the booting time is really slow.

Link: https://face-emo-recog1.herokuapp.com/

3.9.2 AWS EC2 Instance

Challenges:

Amazon Web Services provide top notch virtual computing services. It would not be considered as challenging but there was an issue. The external URL generated by Streamlit was an unsecure ip address which is typed along with the port to access the application, due to which Google Chrome wasn’t allowing media access. It had to be manually excluded to allow all access to start any media device or access the webcam.

The future work would include handling this issue by connecting a domain and ssl certificate to the web application.

Link: http://54.84.63.103:8501/

4. Conclusion:

4.1 Conclusion and Recommendations:

- Face Emotion Recognition is a crucial application of deep learning algorithms which can be extended to every industry.
- Future work in relation to this project can include tracking and analyzing the emotions of the students. For example If

a student is continuously predicted to be sad for a class of an hour, he/she could be flagged and a report of all the students could be generated at the end of the lecture for better analysis and further customized lesson plans.

- Another important point to conclude is CNN models could achieve extraordinary results if appropriate and good amount of training data is provided. For example, for this particular case, the training data should include images of students while studying.
- The model gave 71% accuracy for training data and 66% for validation data. On the other hand it gave 68% accuracy for the test.

4.2 Challenges:

- Image dataset was large to handle in terms of storage and processing.
- Computing time and Google Colaboratory limits, to avoid crashing.
- Model Architecture
- Deployment

5. References:

- Machine Learning Mastery
- GeeksforGeeks
- Analytics Vidhya Blogs
- Towards Data Science Blogs
- Built in Data Science Blogs
- Scikit- Learn Org
- Baeldung.com
- https://edps.europa.eu/system/files/2021-05/21-05-26_techdispatch-facial-emotion-recognition_ref_en.pdf