Facial Expression Recognition with Deep Learning

Improving on the State of the Art and Applying to the Real World

(Observant Force Group)

Shubham Deshmukh – Shubhamdeshmukh278@gmail.com

Saurabh Yadav - saurabhyd423@gmail.com

***Abstract*—one of the most universal ways that people communicate is through facial expressions. In this paper, we take a deep dive, implementing multiple deep learning models for facial expression recognition (FER). Our goals are twofold: we aim not only to maximize accuracy, but also to apply our results to the real-world. By leveraging numerous techniques from recent research, we demonstrate a state-of-the-art 69% accuracy on the FER2013 test set, outperforming all existing publications. Additionally, we showcase a mobile web app which runs our FER models on-device in real time.**

I. Introduction

Facial expressions 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 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 (ex Zoom) where it's not possible for medium scale classes (25-50) to see all students and assess 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 analysed using deep learning algorithms. Deep learning backed systems not only solve 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.

II. Problem Statement

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.We will solve this challenge by applying deep learning algorithms to live video data. The solution to this problem is by recognizing facial emotions.

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.Deep learning is an AI facial expression recognition function that works like the human brain by processing data and developing patterns used for detecting objects and even in decision making. It is a subset of machine learning and artificial intelligence technology.

III. DATASETS

FER2013 Dataset

FER2013 is a well-studied dataset and has been used in ICML competitions and several research papers. It is one of the more challenging datasets with human-level accuracy only at 65±5% and the highest performing published works achieving 75.2% test accuracy. Easily downloadable on Kaggle, the dataset's 35,887 contained images are normalized to 48x48 pixels in grayscale. FER2013 is, however, not a balanced dataset, as it contains images of 7 facial expressions, with distributions of Angry (4,953), Disgust (547), Fear (5,121), Happy (8,989), Sad (6,077), Surprise (4,002), and Neutral (6,198)

Figure 1: Images from each emotion class in the FER2013 dataset.

IV. Data Preprocessing

Auxiliary Data & Data Preparation:
Although several FER datasets are available online, they vary widely in image size, color, and format, as well as labeling and directory structure. We addressed these differences by simply partitioning all input datasets into 7 directories (one for each class). During training, we loaded images in batches from disk (to avoid memory overflow) and utilized Keras data generators to automatically resize and format the images.

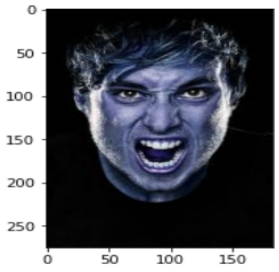
Data Augmentation
We researched and experimented with commonly used techniques in existing FER papers and achieved our best results with horizontal mirroring, ±10 degree rotations, ±10% image zooms, and ±10% horizontal/vertical shifting.

V. Modeling

MODELS

I. Deepface Model

Deepface is a facial recognition and attributes analysis framework for python created by the artificial intelligence research group at Facebook in 2015. Keras and Tensorflow inspire this library's core components. Deepface is a lightweight face recognition and facial attribute analysis (age, gender, emotion and race) framework for python.



We imported the image above which looks angry but our model gives us "27 years old white fear Man" this result. To get better results we decided to train our own model.

II. ResNet50 model

Since the FER2013 dataset is quite small and unbalanced, we found that utilizing transfer learning significantly boosted the accuracy of our model.

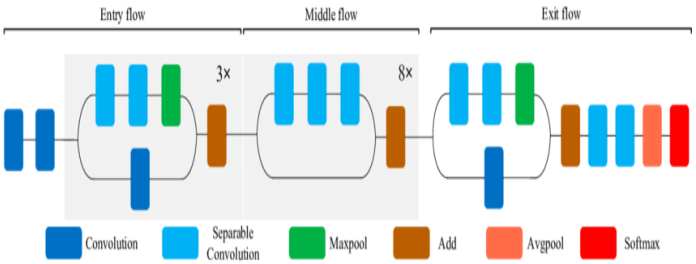
Fine Tuning ResNet50

ResNet50 is the first pre-trained model we explored. ResNet50 is a deep residual network with 50 layers. It is defined in Keras with 175 layers. We replaced the original output layer with one FC layer of size 1000 and a softmax output layer of 7 emotion classes. We used Adam as our optimizer after training for 50 epochs using Adam and a batch size of 785, we achieved 63.11% accuracy on the test set and 67% on the train set. There is much less over-fitting.
We have taken epochs as 50. Once the threshold is achieved by the model and we further tried to train our model, then it provided unexpected results and its accuracy also decreased. After that, increasing the epoch would also not help. Hence, epochs play a very important role in deciding the accuracy of the model, and its value can be decided through trial and error.

III. Xception Model

Xception architecture is a linear stack of depth wise separable convolution layers with residual connections. This makes the architecture very easy to define and modify; it takes only 30 to 40 lines of code using a high level library such as Keras or Tensorflow not unlike an architecture such as VGG-16, but rather un- like architectures such as Inception V2 or V3 which are far more complex to define. An open-source implementation of Xception using Keras and Tensorflow is provided as part of the Keras Applications module2, under the MIT license.

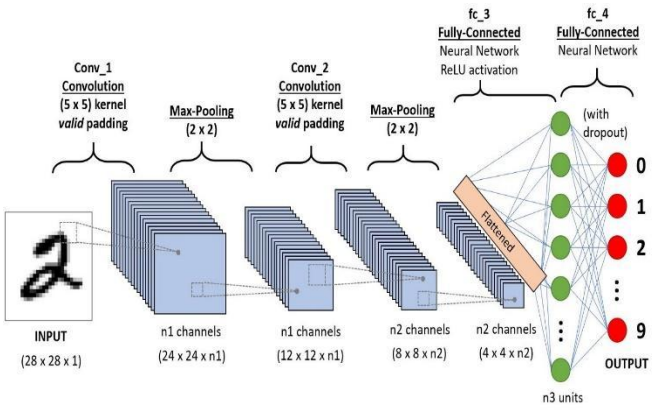
We used Adam as our optimizer after training for 70 epochs using Adam and a batch size of 785, we achieved 64% accuracy on the test set.



The above image shows the final infrastructure of the Xception model. A fully connected neural layer that contains residual depth wise separable convolution where each convolution followed by batch normalization and Relu activation function. The last layer applies a global average pooling and softmax activation function to produce prediction.

IV. CNN Model

A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.



We designed the CNN through which we passed our features to train the model and eventually test it using the test features. To construct CNN we have used a combination of several different functions such as sequential, Conv (2D), Batch Normalization, Maxpooling2D, Relu activation function, Dropout, Dense and finally softmax function.

We used Adam as our optimizer after training for 70 epochs using Adam with minimum learning rate 0.00001 and a batch size of 785, we achieved 69 % accuracy on the test set and 74% as train accuracy.

Results / Discussion

Accuracy-Driven Models

Table 1 shows the accuracies our best models achieved on the FER2013 private test dataset.

Model	Train Score	Test Score
Deep Face	NA	NA
Resnet50	68%	62%
CNN	74%	69%
Xception	72%	64%

Weighted Average	Resnet50	CNN	Xception
Precision	63%	69%	67%
Recall	63%	69%	67%
F1 score	63%	69%	67%

Most of the publications which achieved state-of-the-art accuracies on FER2013 utilized auxiliary training data.

Table 1 demonstrates our accuracy gains from employing auxiliary data with care taken to avoid dataset bias. It also depicts our success in implementing class weighting, which significantly increased accuracies on frequently misclassified emotions.

Table 2 demonstrates the weighted average of precision, Recall, F1 score parameters of the confusion matrix. From the table we can conclude that the best F1 score was obtained by CNN model.

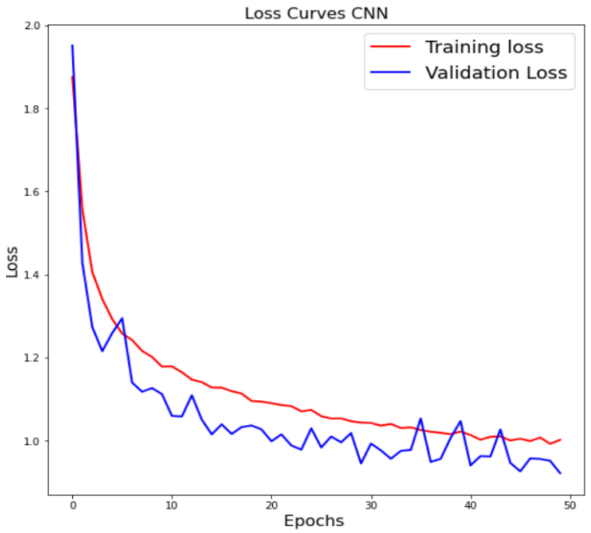
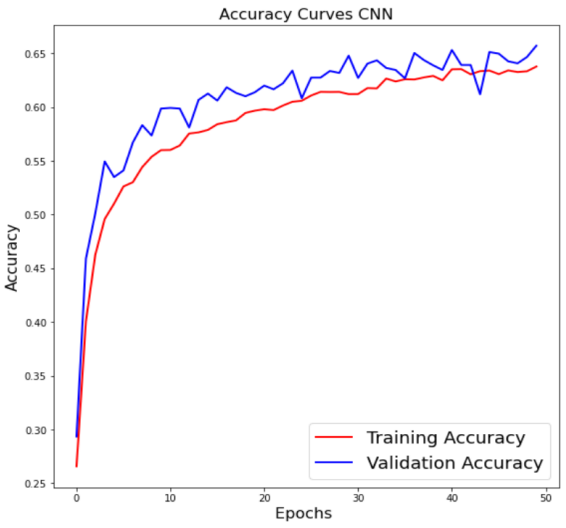


Figure 5: Learning curve for CNN model.



Confusion Matrix

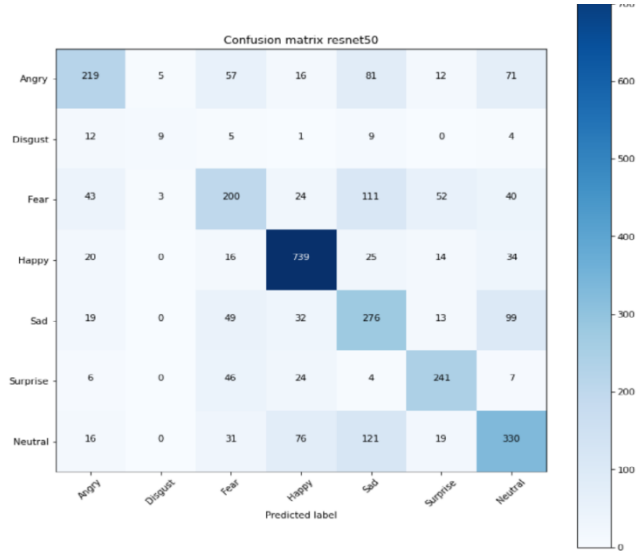


Figure 6: Confusion matrix. Of Resmet_50

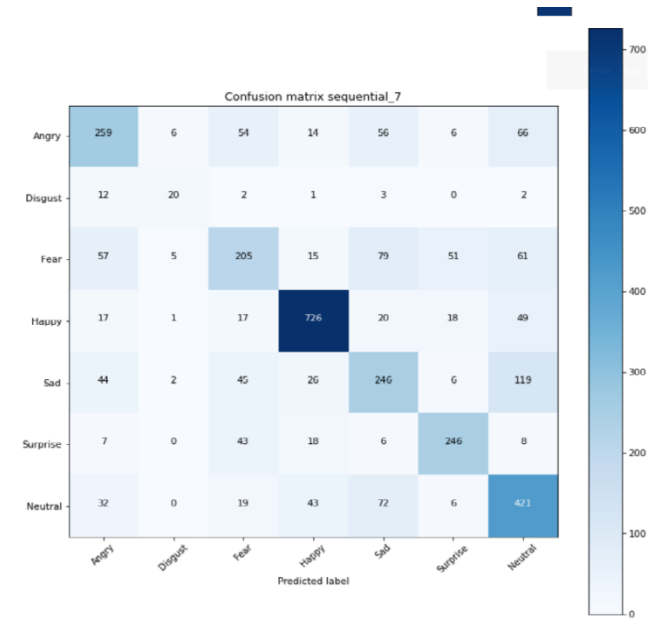


Figure 6: Confusion matrix. Of Custom CNN

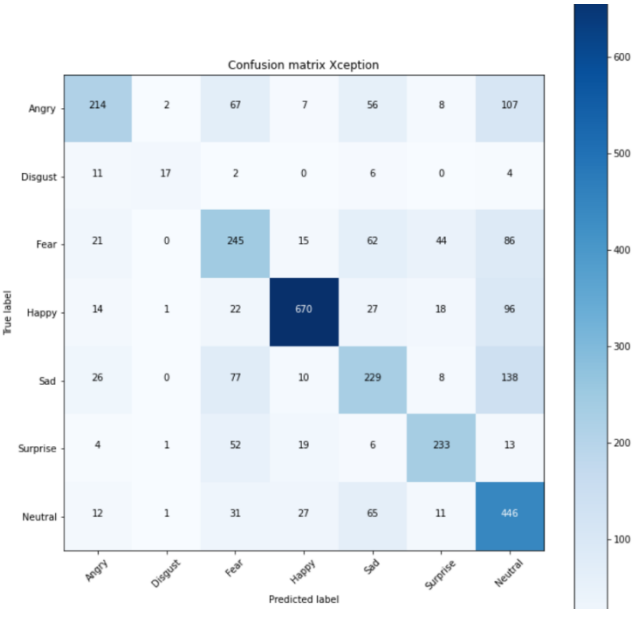


Figure 6: Confusion matrix. Of Xception

For error analysis, we targeted confusion matrix cells with high misclassifications. One interesting example was an image labeled fear that was classified by our model as angry, fear, and surprise, similar to miss predictions by humans on the same image.

We also investigated the high misclassification of angry images as surprise, where one of the subjects was misclassified on all of his surprise images.

Real time local video face emotion detection

We created patterns for detecting and predicting single faces as well as multiple faces using OpenCV video capture in a local webcam.

Some of the detected face emotion are as follows:

VII. Web APP

Rather than take a purely theoretical approach, we thought it would be challenging and novel to apply our work to the real world by developing a web app to run our model on-device in real-time. We deployed the app in Heroku. If you see in the starting section of GitHub repo you see that all the requirement files are there for creating an app on Heroku of name “face-emotion-recognition-off”. But due to high slug size the buffering takes time so we have run our app working on local and it ran properly and app is also fine also we’ve included video on GitHub repo.

HerokuLink: -

https://face-emotion-recognition-ofg.herokuapp.com/

VIII. Challenges

1) Large Image Dataset to handle.
2) Couldn't connect the GPU with the Jupyter Notebook.
3) Tried creating a lot of models till I found the best one.
4) Continuous Runtime and RAM Crash due to large dataset.
5) Carefully tuned hyper parameters.

IX. CONCLUSION

When we started this project, we had two goals, namely, to achieve the highest accuracy and to apply FOR models to the real world. We explored several models including shallow CNNs, Deepface, ResNet50, and Xception. To alleviate FER2013's inherent class imbalance, we employed class weights, data augmentation, and auxiliary datasets. By ensemble seven models we achieved 77% training accuracy and 69% accuracy for CNN model. We also found through network interpretability that our models learned to focus on relevant facial features for emotion detection.

Additionally, we demonstrated that FER models could be applied in the real world by developing a web application with real-time recognition speeds. It was an interesting project and we learned a lot from it.

X. CODE

A poster and video showcasing our work can also be found in our GitHub repository.

https://github.com/ShubhamDeshmukh27/Face-emotion-detection-deep-learning-project.git

https://github.com/saurabh423/Face_Emotion_Recognition.git

CONTRIBUTIONS:

Saurabh Yadav

- Auxiliary data and dataset preprocessing
- Mobile web app development and deployment also done model tuning
- Build Exception and Custom CNN models
- Error analysis and network interpretability

Shubham Deshmukh

- Build DeepFace model
- Transfer learning models(Resnet_50)
- Data augmentation
- Hyperparameter tuning