**Facial Expression Recognition System Using Pytorch, Fast.ai and Tensorflow**

Sharon I1, Ojeifo O2, Praise T 3, Kayode A4, Ayodele A.5

AI SATURDAY DEEP LEARNING COHORT GROUP 1 FINAL PROJECT REPORT SUMMARY

*GITHUB REPO:* [*https://github.com/AI6DLProject*](https://github.com/AI6DLProject)

**Case Study**

A certain retail company desperately wants to retain existing customers than acquire new ones via advertising and other methods. After performing analysis on existing customers, the data analyst team reported that poor customer service is often the root of losing loyal customers. Your company (an AI consulting firm) proposes to build a facial recognition system capable of predicting the state/emotion of customers, so as to notify employees around the store to immediately assist the customers with questions and concerns about a product.

**Introduction**

For most firms, especially growing SMEs, customer retention is one of the major struggles, therefore, causing businesses to seek for smarter approaches to promoting good relationships with first time and regular customers. The impact of AI has gone beyond some distant applications and are now applied in many sectors because of how well it provides solutions to problems in most cases. Taking for instance where some mobile apps ask users for a rate – either in form of number of stars or using some emojis (the happy, neutral and sad faces). This at times, help businesses improve the performance of their apps if most people rated them between fair to poor. But this method does not cover the scope of their offline services to offline customers. There is also need to know what people who fall into this category think about the business customers’ service. Rather than giving out survey forms which is cliché, and often times seen as a burden to fill, if there were to be a device that can automatically sense emotions, it will perhaps be a better solution.

**Aim**

To build a model that can accurately predict customers emotions from their facial expressions. If deployed, this solution should be successful by connecting this model to a monitoring camera in the business place to detect the expressions on random customers the camera captures. The result will be transmitted to the operations manager in charge. This approach should cause the following:

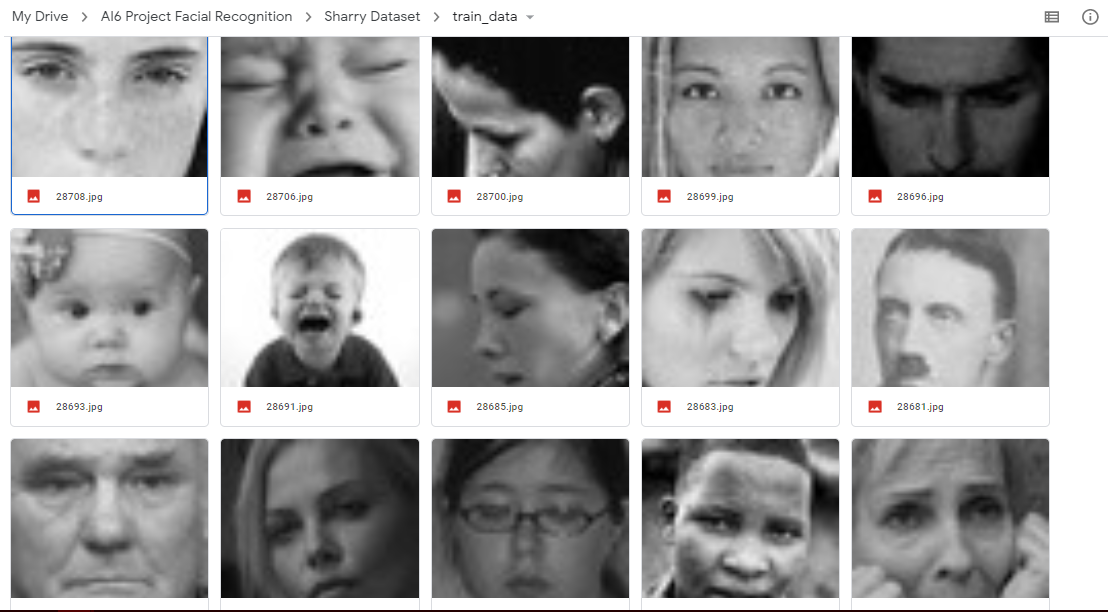
* The manager assigning a less busy staff to whichever customer(s) the model predicted as being sad
* Making the business investigate into their operational system for improvements if there are frequent occurrences of customers’ displeasure.

**Objectives**

* Gather enough data on facial expressions
* Organise the data into three classes (**Sad**, **Happy**, **Neutral**) for classification.
* Build a model and achieve a high accuracy.

**Tools and Methods**

The project was done on Google Colab with the following deep learning frameworks: PyTorch, fastai, and TensorFlow. Out of which PyTorch produced a model with the highest validation accuracy of 98% and Test accuracy of 76%. The link to this notebook is right under the project topic on the first page.



*Fig 1.0*

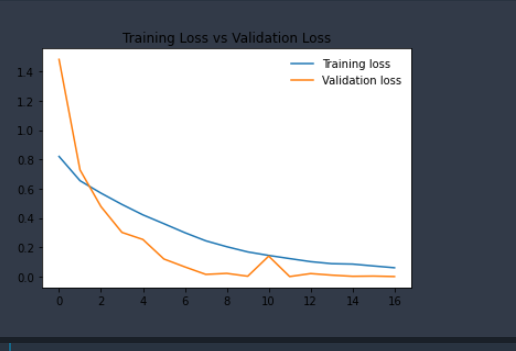
Kaggle’s facial expression recognition dataset (fig 1.0) had tons of images that were valuable to the project idea. Therefore they were downloaded on Colab to the Google drive using *!wget.* These images were further cleaned from the drive by deleting bad images and properly classifying the misclassified ones. We also went further to reduce the number of classes which were seven from Kaggle, to the three classes which are the focus of the project. After doing this, we defined the path on Colab which was where these images were located on the Drive, so as to be able to convert them to a csv file and read them using *pd.read\_csv*(). Since we were going to work with models that understand only numerical values, the image classes were labelled as Happy: 0, Neutral: 1, and Sad: 2. At this point we proceeded to loading the training and testing data using dataloader. Prior to this, some of the hyperparameters, as the validation size, batch size, number of workers and transforms had been defined.

When building models, it is important to try out different models to find the one that performs best. Because of this, it was vital to explore the pretrained models, such as se\_resnext50\_32x4d, se\_resnext101\_32x4d, resnet50, senet154 etc.

**Results**

During this process on PyTorch, it was discovered that se\_resnext50\_32x4d performed best by giving us the accuracy aforementioned with the following hyperparameters values:

* num\_workers = 0
* batch\_size = 4
* valid\_size = 0.15
* epochs = 20
* optimizer = Adam(lr=1e-5)
* scheduler = ReduceLROnPlateau(mode="min", patience=5, factor=0.3, verbose=True)
* criterion = CrossEntropyLoss()



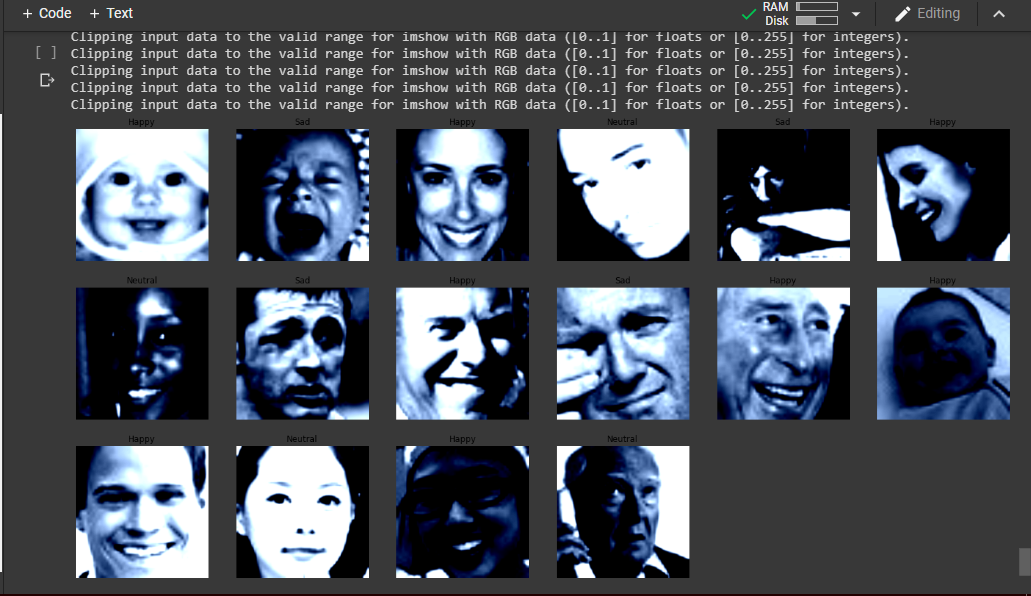
*Fig 2.0 Graph plot of Validation and Training Loss*

Those values were not automatically selected; they had to be altered differently in order to arrive at a high accuracy, yet avoiding over-fitting. (Fig 2.0) shows a graph displaying the training loss plotted against the validation loss after the model was trained. At the end of the end of the training process, we arrived at a validation accuracy of 98% and validation loss of 0.0420 (Fig 2.1). The model was tested on the test data with a Test Accuracy of 76% and Test loss of 1.1272 (Fig 2.2), whose outcome was shown in the notebook. From the visualizations, one can say that the model did well by accurately classifying these images (Fig 2.3).



*Fig 2.1 Validation Accuracy*

*Fig 2.2 Test Accuracy*



*Fig 2.3 Test Validation Samples*

**Limitations and Conclusion**

As seen from the sample data in the notebook, the model was built on a biased dataset that contained thousands of whites than black faces, hence it is most likely going to perform better on white people than blacks whereas the application of this model is projected to the latter. Due to the unavailability of a better dataset, we still had to work with what we could find.

Conclusively, using AI in the business space, should lead to substantial improvements and promote productive outcomes. Therefore, this approach if deployed, should support the retail business as stated in our case study to evaluate customers’ feelings without literally asking them, and using it to promote good customer service. By so doing, the business can retain their customers and be impressive as well to their first-time customers.