Yelp Restaurant Multiclass Classification Using Images From Reviews

Samantha Chu

samantha.chu@marylandsmith.umd.edu

Jenil Kansara

jenil.kansara@marylandsmith.umd.edu

Shweta Malla

shweta.malla@marylandsmith.umd.edu

PROJECT MOTIVATION

The motivation of this project is to help Yelp tag their images based on contextual information and help automatically enhance user reviews to increase credibility. Due to the three dimensionality of images (height, width, color) traditional neural networks are unable to process these images, so we decided to use Convolutional Neural Networks.

DATA

We used the data provided by Yelp from their crowdsourced reviews page. The dataset includes images (with photos IDs), the business IDs, and the multiple labels attached to those images.

OBJECTIVE

We worked with a user-submitted image dataset from Yelp with 9 possible labels. We worked on this problem, which is a multi-instance, and multi-label classification problem, using Convolutional Neural Network (CNN) in Python using keras deep learning library.

EVALUATION METRICS

To check the validity of our model, we decided to use the AUC metric along with the F-score. The higher AUC and F-score determines a good model. In our case, the cost of a false negative is high. If the cost of false negative is different than that of false positives we look at the F-score to determine model feasibility.

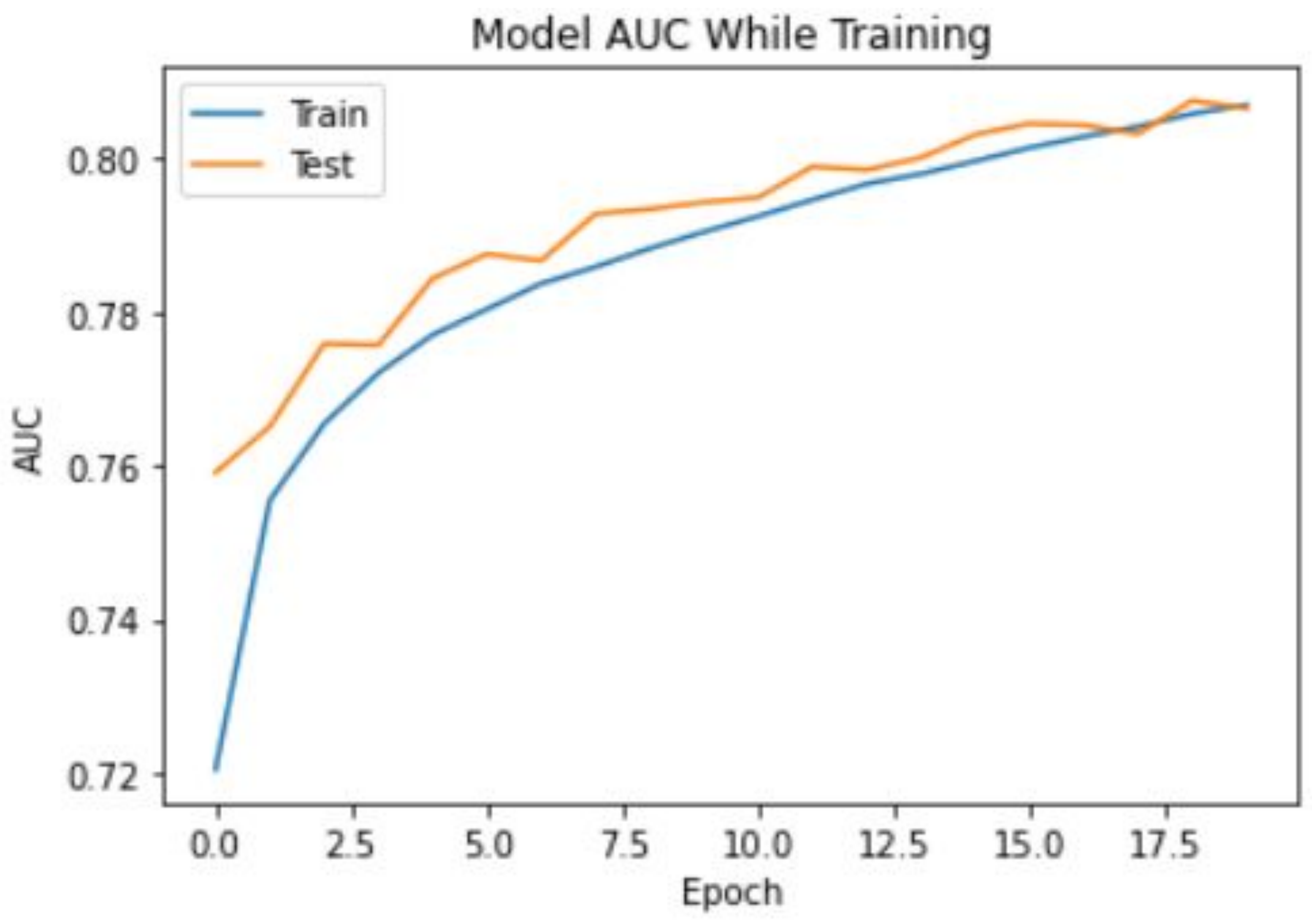


Figure 1. ROC Curve
With an AUC (Area Under Curve) value of **0.80**. This model also had an F1 score of **0.739.**

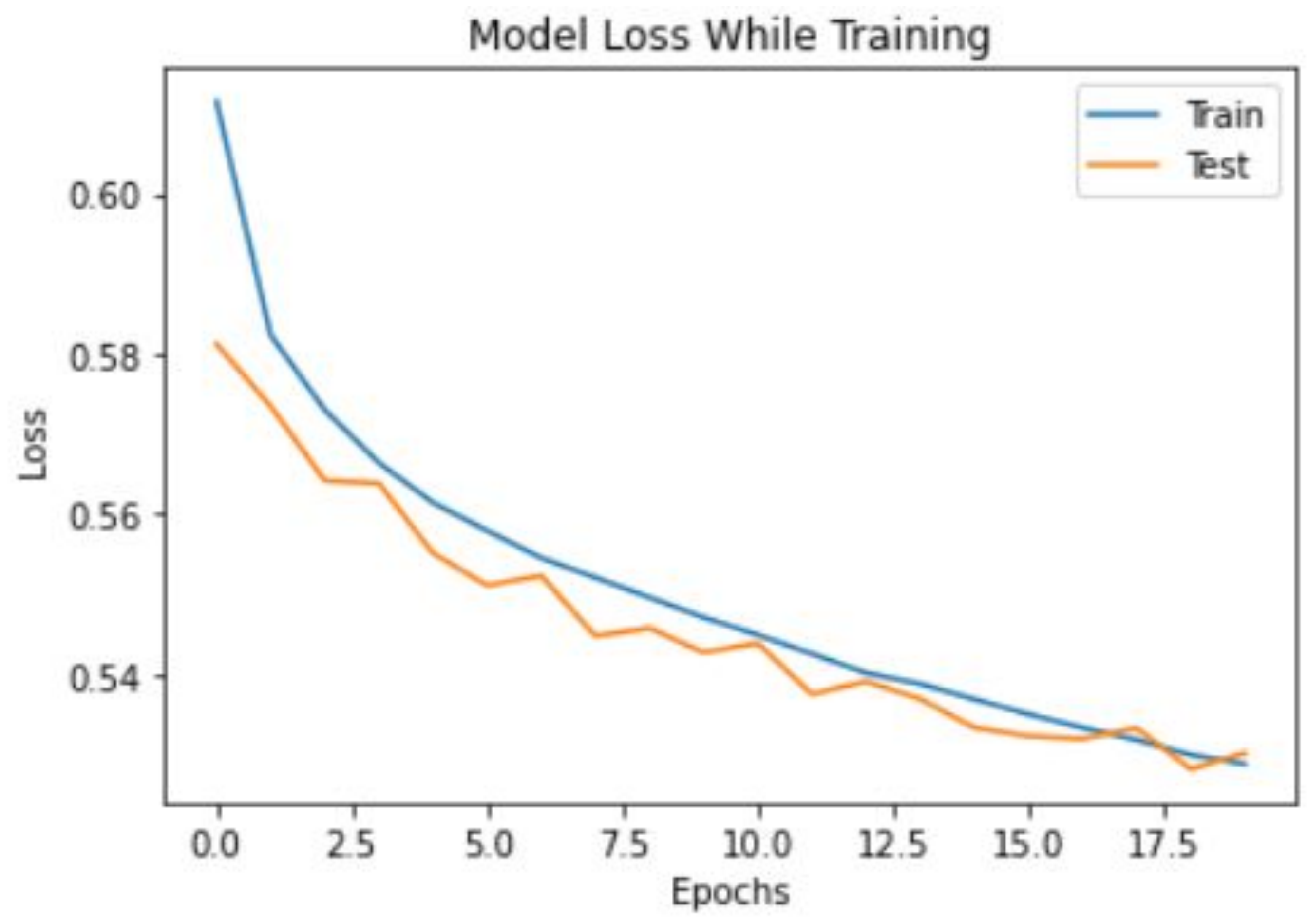


Figure 2. Model Loss Curve
After each epoch or iteration, the loss was calculated and graphed, in the above chart.

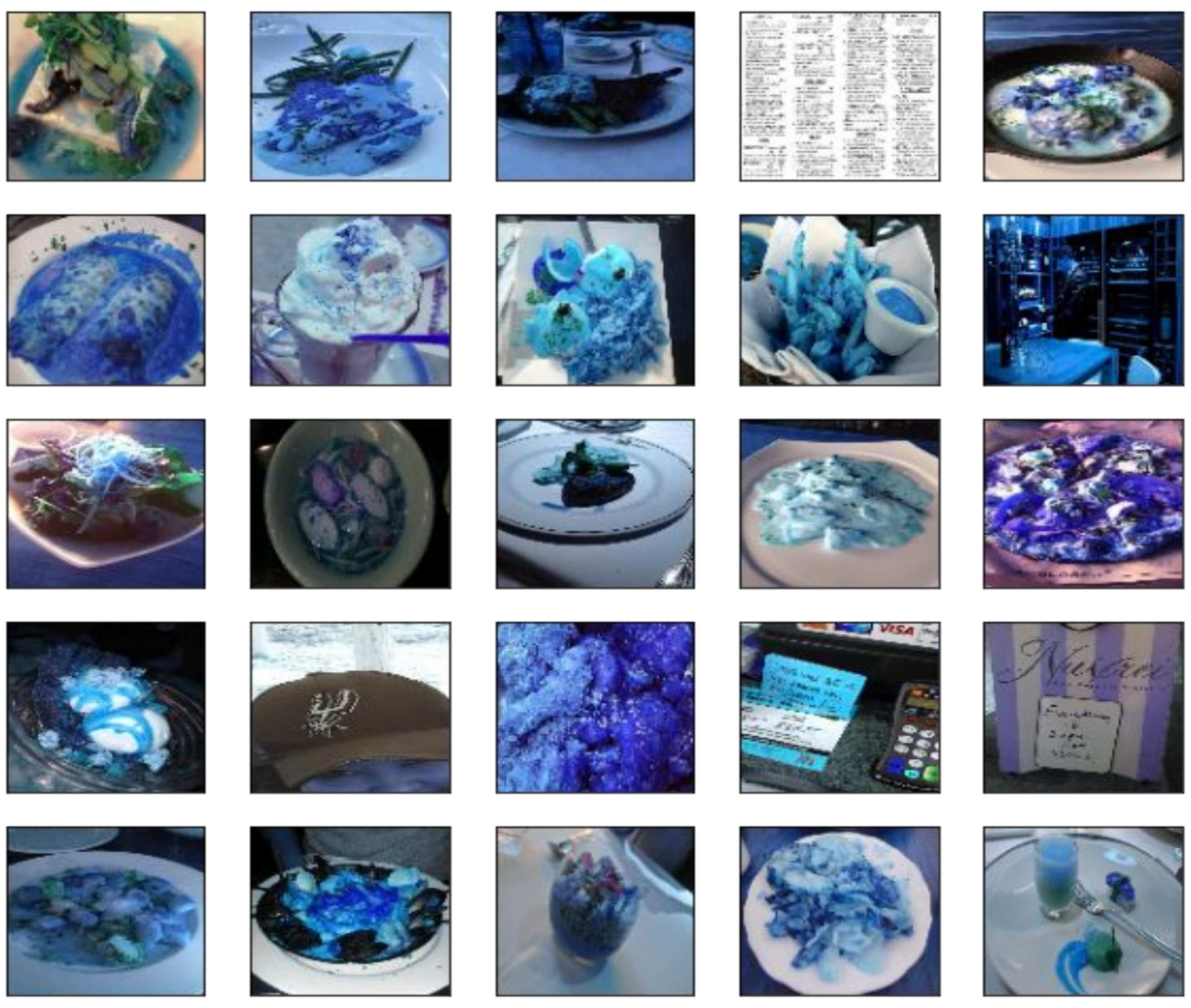


Figure 3. Training images after re-scaling and normalization

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

Figure 4. F1-Score Calculation Formula

METHODS

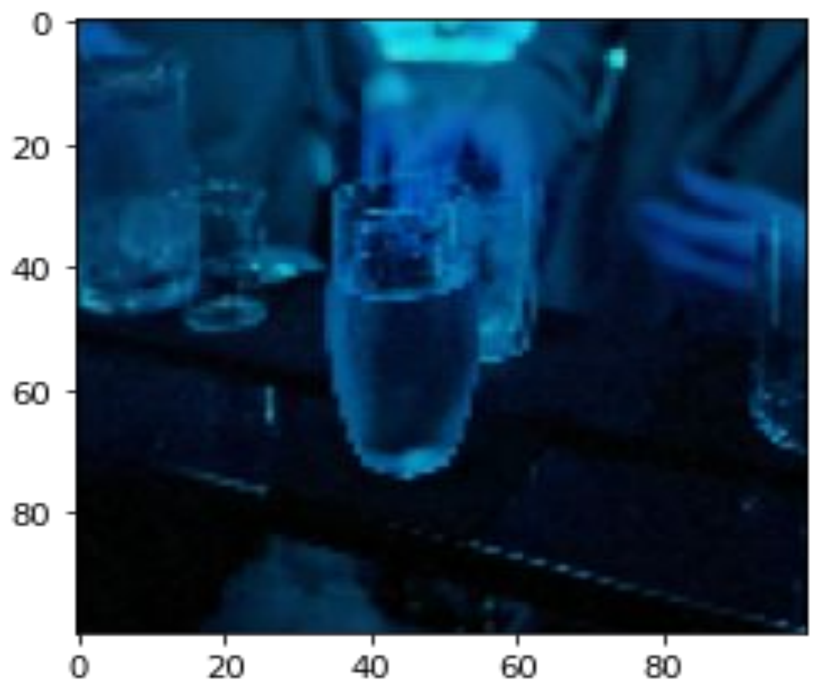
To build our CNN, the initial step was to pre-process our data, normalize and rescale our images. We used ReLU as the activation function and implemented hidden layers and added dropout as well to avoid overfitting. We were only able to use 150,000 images due to lack of computational power.

CONCLUSIONS & FUTURE SCOPE

Our CNN model does a great job for predicting labels for images.
We were able to tune the hyper parameters to achieve a low loss and a high AUC value. As the number of epochs increases the loss decreases. The losses are in line with each other, which shows that our model does not overfit or underfit.
For further research, this model can be used as a computer vision transfer model for predicting the labels of tourist attraction images.

RESULTS OUTPUT

To check the accuracy of our model, we tested it on an image of a bar with a drink on the table. Our model accurately labelled the image as 'Has alcohol' and 'Has table service'




```
finalPrediction
['Has alchohol', 'Has Table Service']
```

Figure 4. Restaurant image and predicted output