San José State
UNIVERSITY

Yelp Restaurant Image Classification
CMPE-255(02) Data Mining Project

Group 07 - Project Report

TEAM – Spartans

Navoday Singh Tomar (12364326)

Rohit Shrishrimal (012435111)

Ishaan Singh (12446746)

Guided By: Prof. Gheorghi Guzun

1 Introduction and Background

Today we live in a society where each and every person is connected to each other one or the other way via internet.
Uploading food pictures, sharing what they eat with friends, family & others have become a common habit of people around the world. Whenever a user uploads a picture he/she had to manually tag the type of cuisine it is. In an age of food selfies and photo-centric social storytelling, it may be no surprise to hear that Yelp's users upload an enormous number of photos every day alongside their written reviews & type of cuisine it is.

1.1 Motivation & Objective

In this project, we will build a model that automatically tags restaurants with multiple labels of the cuisine type using a dataset of user-submitted photos. Currently, restaurant labels are manually selected by Yelp users when they submit a review. Selecting the labels is optional, leaving some restaurants un- or only partially-categorized. With this data mining technique, the problem for un- or only partially-categorized restaurants will be solved. Also, users will have great experience where they don't have to type the cuisine type.

1.2 Market Review

In this tech Savvy world there is lots & lots of competition among companies to out run each other. All that matters is customer satisfaction & great user experience. Companies are getting competitive, adding new and innovative feature to have great user experience. Having these kinds of features will attract customers as well as will help improve the business of restaurants as well.

2. System design & Implementation details

2.1 Algorithm Used

1. Get the photo dataset and business dataset from yelp data repository.
2. Load the photos and the business features.
3. Extract only those photos from the json data which are tagged with label - "Food".

4. Use t-SNE/Truncated SVD/PCA and compare the results for feature extraction and dimensionality reduction on the input data.
5. Try dimensionality reduction before and after applying the variations to input set of images.
6. Create the different variations of each and every image and similarly keep track of the tag associated with original image with the newly generated images.
7. Pass the input data to the 5 layer of network i.e. 3 layers of Convolution feeding output of first to second and so on, flattening the output and then finally feeding it to fully connected layer.
8. The resulting Convolution Neural Network is used to train the model based on random weights and biases generated by the functions defined for it using the standard deviation according to resolution of images. 3000 iterations are run to finalize the weights and get an accurate model.
9. This model is then used to classify any image fed to it as input by the user.
10. Based on the image, a tag is identified, which in turn relates the image to a business id(Restaurant). This data is used to return a new set of tags and then using these tags we find the type of cuisine in the business.

2.2 Technology & Tools Used

- Numpy - For handling the datasets (pip install numpy)
- Pandas - For handling the datasets (pip install pandas)
- Scikit Learn - To use classification algorithms like SVM
- Python
- H5Py - To store the features extracted from CNN (pip install h5py)
- Caffe - To extract features from the images
- TensorFlow
- Opencv
- Imutils

2.3 Architecture

Following diagram represents the architecture of the system.

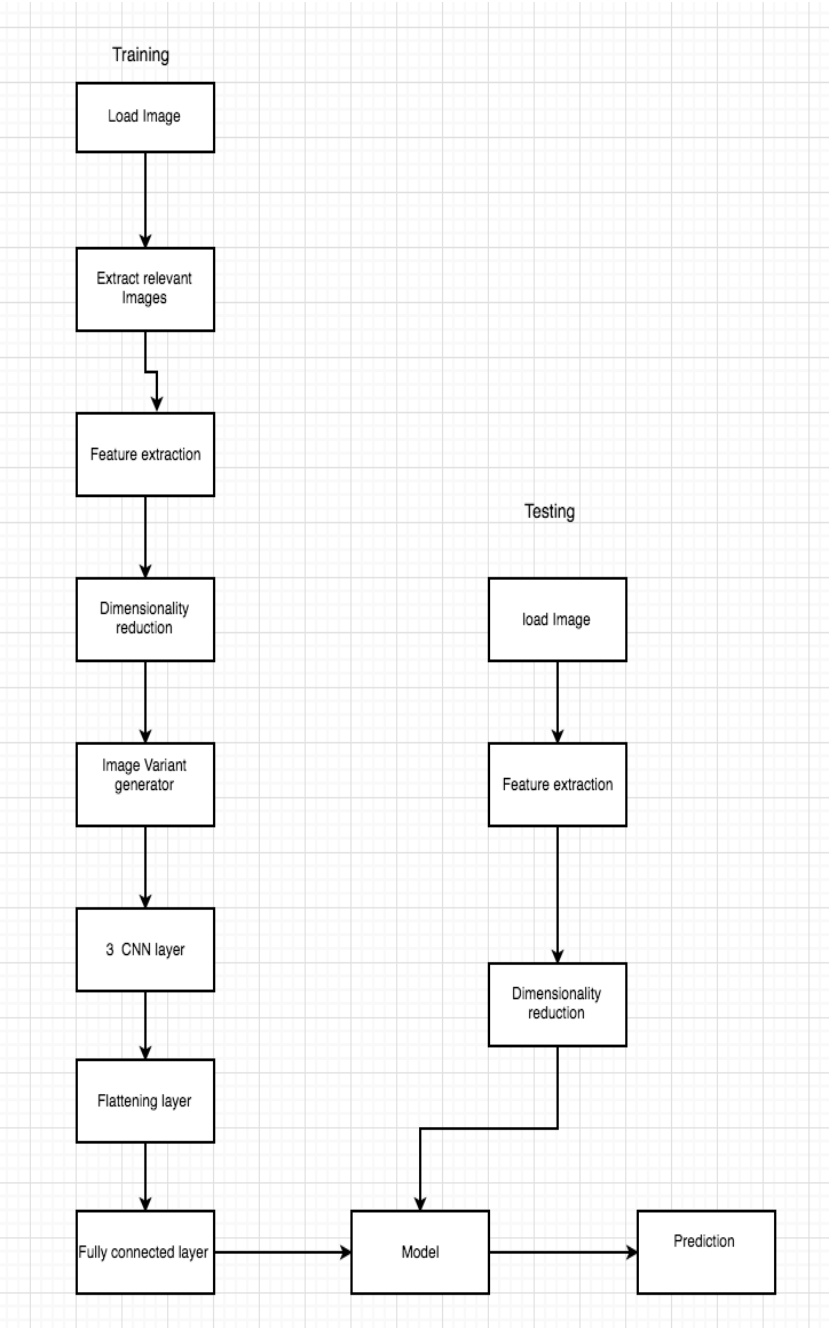
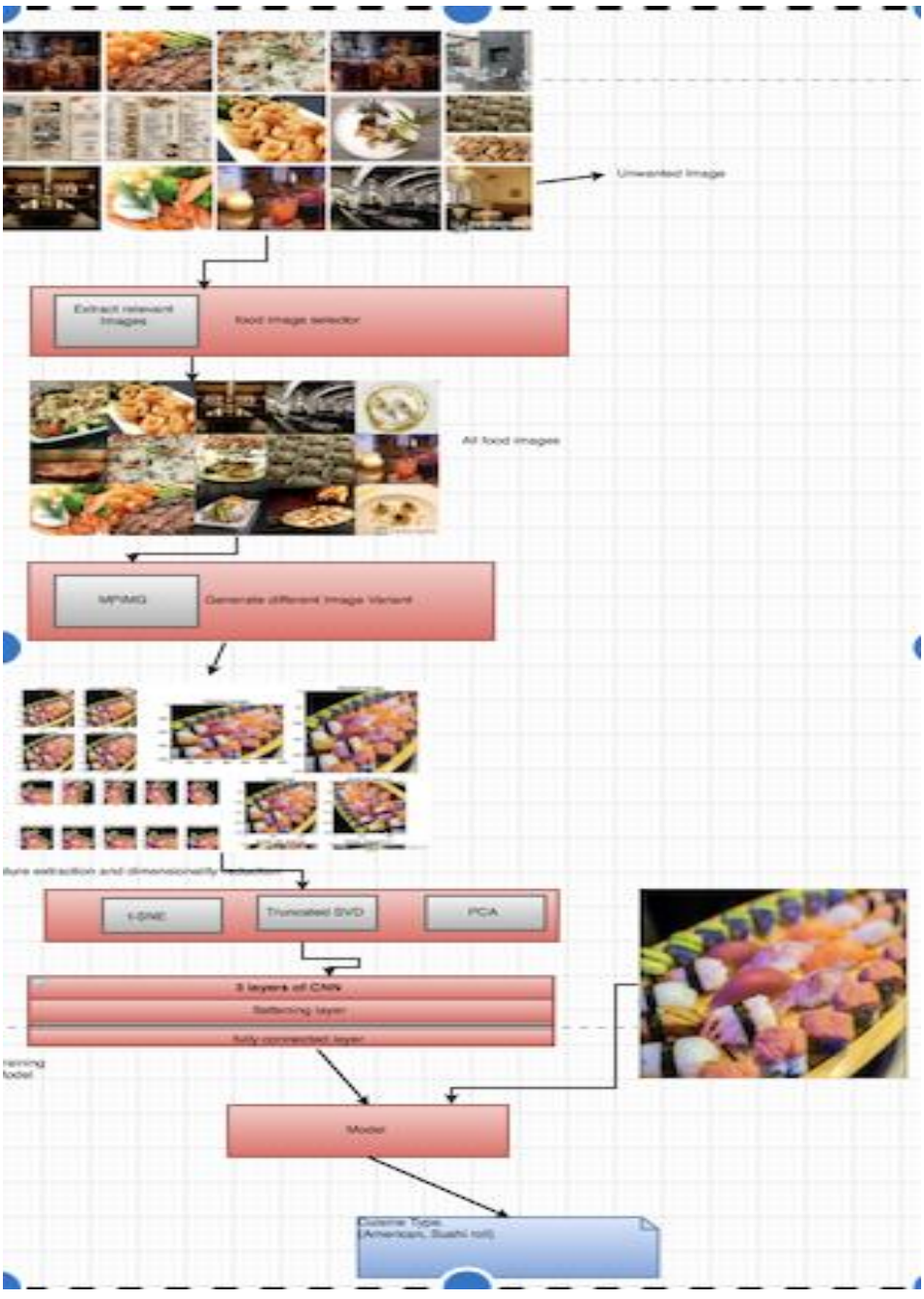


Fig: 1.1

2.4 Data Flow & System Design

Following are the components in our System:

a) Load image.
b) Extract relevant images.
c) Feature extraction & dimensionality reduction.
d) Image Variant Generator.
e) 3 CNN layer.
f) Flattening layer.
g) Fully connected layer.
h) Model
i) Prediction



3. Experiments / Proof of concept evaluation

3.1 Dataset Used

The photos dataset contained 206949 images in total including the restaurants, interior, outside, food, ambience etc. labels for them.We extracted only the photos labelled as “Food” for our model. The number of such images is 132354. Extracted data contains - Photo_id, Business_ID. This Business ID is then utilized to extract business related data e.g. categories from the business dataset. There is a JSON file accompanying the images which we parsed to get the id(s) of images labelled as food, then used those ids to read the image files(.jpg) and extract RGB, Hist data.

3.2 Graphs

3.2.1

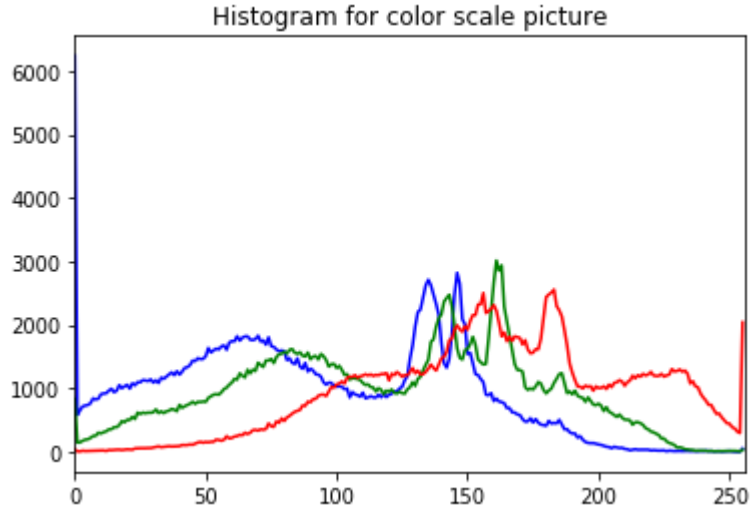


Fig 3.1

3.2.2

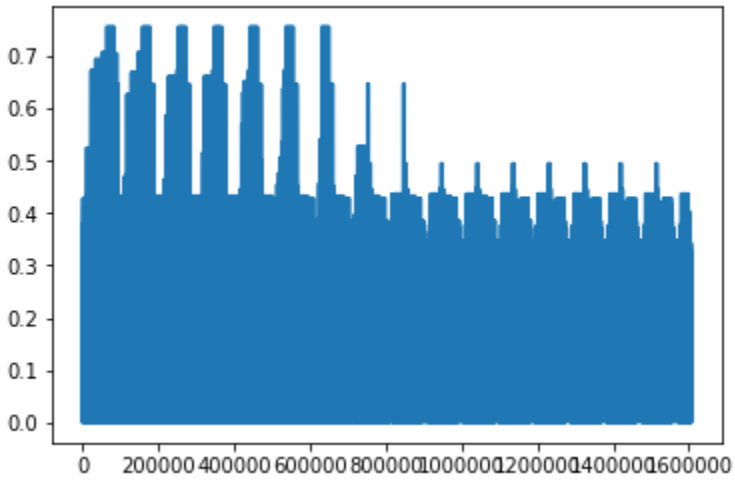


Fig 3.2

3.3.3

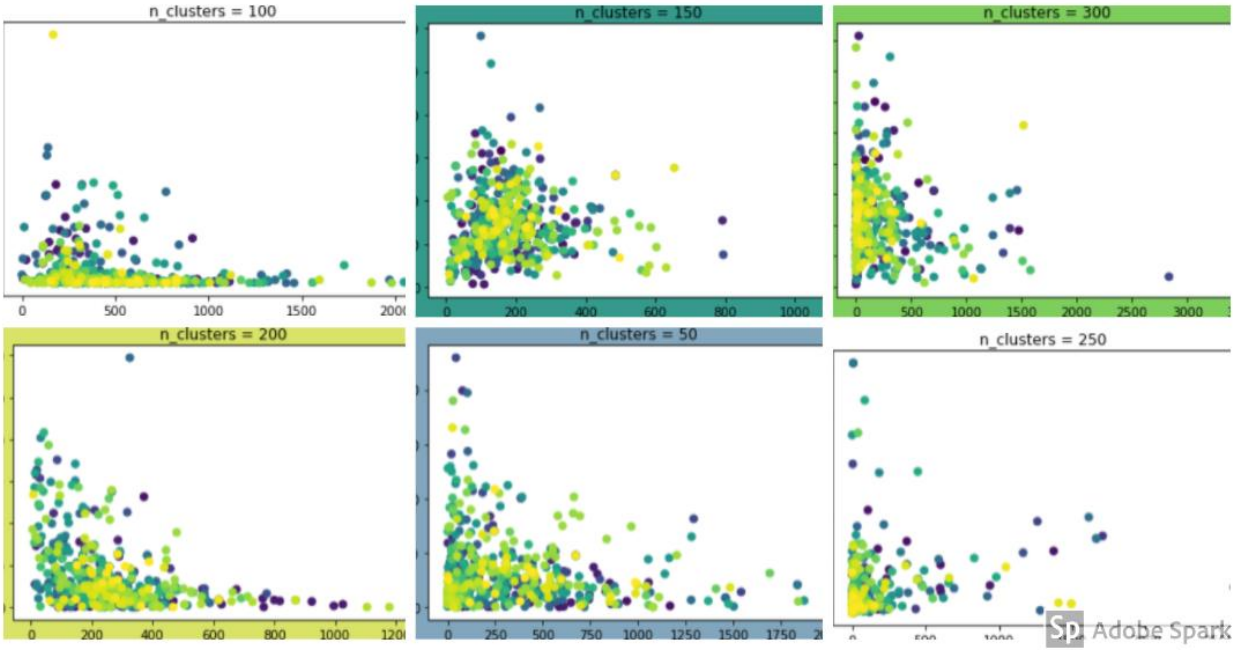


Fig 3.3

3.3 Analysis of result

Using XGBoost we got the accuracy of 61.80% whereas using Tensorflow we were able to reduce the loss to 0.15%. With found that with high dimension data like food images Tensorflow works better compared to XGBoost.

```
y_pred_test = selection_model.predict(select_X_test)
accuracy = accuracy_score(photo_id_to_business[s:n], y_pred_test)
print("n=%d, Accuracy: %.2f%%" % (select_X_test.shape[1], accuracy*100.0))
```

```
n=22, Accuracy: 61.80%
```

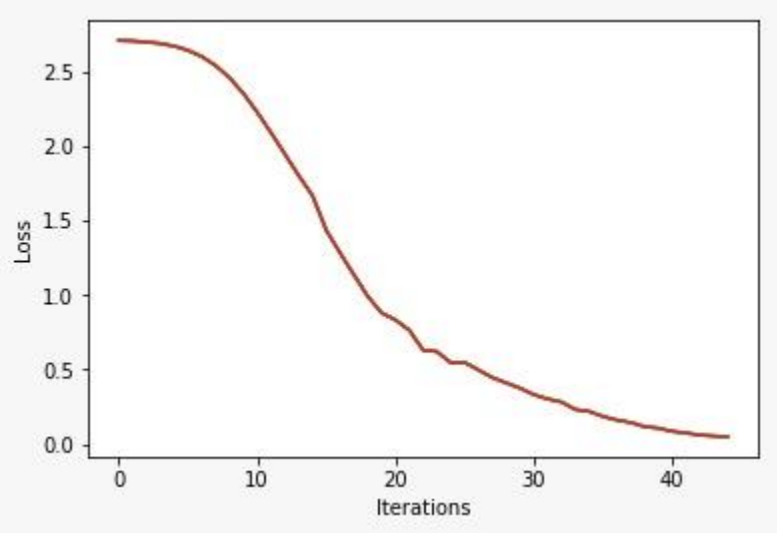


Fig 3.4 Loss vs Iterations for Tensorflow

4. Discussion & conclusion

4.1 Decision Made

As Image classification with neural networks is a new area where we were working on we faced many road blocks. Many Decision we took during the course of this project. Few of them are:

1. As the dataset is huge and we don't have the resources to train the model with 4 million data. So, we selected 70% of the data to train the model.
2. User can upload any type of food image, image with different angle or the image can have distortion, or the image can a partial image & so.

The possibilities were enormous. We created 29 variants of a single image.

3. For testing we can't test it on whole data set, because the resources available on our laptop doesn't allow this much processing and HPC in our college doesn't support Tensorflow.
4. Dimensions of the images are huge. Reducing those dimensions was important using t_SNE.
5. Feature extraction was important keeping RGB & Hist model in mind. Generally, the food images varies in colors only.
6. We tried to train the data with XGBoost & Tensorflow. The best result we got is from Tensorflow.

4.2 Difficulties faced

Following are the difficulties faced by us during the course of this project.

1. Training the huge data was one of the big issue. It requires enormous GPU power for this huge data set.
2. Maintaining the respective tags for all the 29 variants of the images created for a single image.
3. Appending the newly created variants with the original image.
4. Testing on all the dataset was impossible with the compute power we have on our laptops. It was working perfectly on random sample we were picking to test.
5. Dimensional vector reduction was difficult.
6. For modeling three layers of CNN, one layer of flattening layer & one fully connected layer. So as the number of layer increased the prediction accuracy increased along with the complexity of the model.

4.3 Things Worked

Currently we are able identify the cuisine type for the random samples we were testing on our model. If we had the high compute power, we could run the model on all the 4 million images and can check the efficiency of the model.

4.4 Things that did not worked

1. We used linear regression & dimensionality reduction to pick HOG features & utilized them, but it did not worked.
2. We tried to classify Images on the bases of their business instead of identifying them independently.

4.5 Conclusion

After trying the Image classification on huge set of Image with XGBoost & Tensorflow, we can say that when the data is enormous Tensorflow works exceptionally well compared to others. Feature selection & Dimensionality reduction is important aspect of data preprocessing. The food images have high dimensions so to extract correct information out of it Tensorflow was the best option amongst the two of them. Among Histogram of Oriented features, RGB features, Normalised Color Histogram, Local binary features & Depth of field features we selected RBG & Normalised Color Histogram features. Complexity increases as we include more features.

5. Project Plan/Task Distribution

The Project was divided according so that each & every one of us contributed equally. Navoday came up with the idea of trying to do it with Tensorflow in parallel with XGBoost so that we can compare both the results. Ishan handle the data manipulation part where it gets tricky because both the methods works on different kind of data. Rohit along with Navoday worked on feature extraction and dimensionality reduction & handled the 5 layers of Model in Tensorflow. All of us contributed equally in report & presentation as well.