YelpR: Photo Classification of Restaurants

Samantha Davuluri
School of Computer and
Information Science Engineering
University of Florida
Email: ndavuluri@ufl.edu

Anirudh Mallem
School of Computer and
Information Science Engineering
University of Florida
Email: anirudhmallem@ufl.edu

***Abstract*—The Yelp competition on kaggle is about predicting business attributes in the form of labels for restaurants, using user submitted photos. Currently, restaurant labels are manually selected by Yelp users when they submit a review or post a picture. Selecting the labels is optional while submitting a review or posting a picture as a result of which some restaurants are left partially-categorized. Additionally, in some cases restaurants are categorized inaccurately. Through this project we propose a deep learning based solution to classify user uploaded images on Yelp. We use various approaches with convolutional neural networks to label businesses with attributes using the photos of the businesses in this task of multiple instance multi-label learning.**
We found that training a Convolution Neural Network for multiple instances together for multiple labels is tough and very computational and resource intensive for the given Yelp Kaggle dataset. On the other hand we found that applying transfer learning technique to solve the classification problem by pipelining a pretrained CNN (AlexNet) on ImageNet database with a multi-label SVM classifier resulted in an F1 score of about close to 0.79. Individual label performance was also analyzed to draw inferences regarding the model behaviour.

***Keywords*—Neural Network, CNN, SVM, transfer learning, F1 score.**

I. Introduction

The Yelp competition on kaggle is about predicting business attribute labels for restaurants using the user submitted photos. Currently, restaurant labels are manually selected by Yelp users when they submit a review. Selecting the labels is optional as a result of which some restaurants are left partially-categorized. In this competition, we are given photos that belong to a business and are asked to predict the business attributes. There are 9 different attributes in this problem which are:

- good_for_lunch
- good_for_dinner
- takes_reservations
- outdoor_seating
- restaurant_is_expensive
- has_alcohol
- has_table_service
- ambience_is_classy
- good_for_kids.

These labels are presently annotated by the Yelp community and our task is to predict these labels for the restaurant purely from the business photos uploaded by the users.
This problem is here is that the labelling needs to be done at a business level, in this case a restaurant. This may seem trivial but the challenge is that each restaurant has varied number of photos associated with it and there is no way to find out which image was responsible for a particular label associated with it. Additionally, this challenge is unique when compared to other image classification challenges as it requires us to classify multiple images together with multiple labels in contrast to classifying a single image.

II. Architecture

A. Faster - RCNN Approach

Our initial approach was to use the Object Recognition Scene classification technique to get an overview of the whole restaurant in terms of the objects in it by analysing all the images under it. Use this information to implement a classifier for labelling the restaurants.

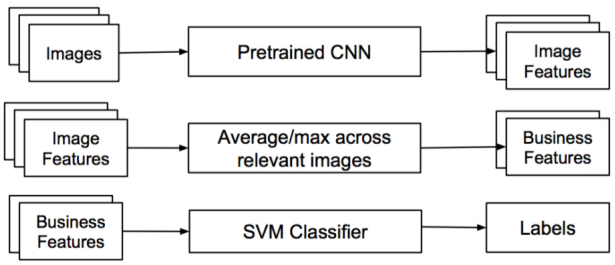
The architecture proposed was similar to that of AlexNet. It would consist mainly of two iterations of two convolution layers followed by a pooling layer. After that we have two fully connected layers followed by a classifier. This classifier would either be a softmax classifier or a linear SVM. We added an extra convolution layer after the pooling at layer 3 in order to extract more mid level features.

We were looking to start from the results delivered by using the Faster-RCNN approach and make changes to the process of region proposing to suit the given dataset by yelp. Faster-RCNN's use a region proposal network [1] as opposed to the traditional selective search used in a normal RCNN. We came across a problem with implementing selective search for the large dataset that we were given as part of yelp. Since here multiple images are linked to multiple business id's we were not sure how to go about implementing a region proposal algorithm.

The yelp dataset is just plain raw images grouped under a business-id and there are around 0.25 million images in the dataset. We realised that this dataset cannot be used to train an RCNN as the training set needs to have the bounding boxes already labelled which was not possible for the Yelp dataset.

B. Multiple Instance Learning Approach

The Yelp classification challenge requires us to classify at a business-id level and not at an image level, i.e, the end result needs to classify a restaurant and not any particular image of a restaurant with the given business attributes. Naturally, the dataset available for training and testing has also been labelled at a business-id level and not to an image level. To summarise, if a particular business-id has multiple images then all the images combined together resulted in the business attributes assigned to the business-id and there is no information as to which image contributed to a particular business attribute for the restaurant. The challenge here is that multiple instances have to be learned with multiple labels. We wanted to simplify this problem to learning one instance with multiple labels and came up with a new transfer learning approach by using a pretrained CNN to extract features and then use these features to train a custom classifier to do the labelling.



We would first train a commonly used CNN like AlexNet or CaffeNet using the ImageNet database and then use this trained network to generate feature vectors for the images available for a particular business-id together in the yelp challenge dataset. This would result in multiple feature vectors for a particular business-id. We intend to reduce these multiple vectors into a single vector by taking the mean across all the feature vectors and result in a single feature vector for a particular business-id. This newly formed feature vector would be a baseline feature vector for the business-id thereby converting our problem into learning about a single instance with multiple labels.

This new problem can be solved using by having a multi-class SVM classifier which can be trained with the consolidated feature vectors created above and help us in classifying our testing dataset. This would work as the consolidated feature vectors generated for each business-id will have all the important features of each image listed under that business-id and help us in training with known algorithms.

We believe this approach of transfer learning will work as the ImageNet challenge has many labels which we can directly relate to the object found in restaurants. For example, ImageNet has multiple labels and images associated to ”Food” which makes a model already trained on the ImageNet database to be good at finding out food in the image. Similarly, the imported model would be good at finding out ”tables” in an image, ”alcohol”/”bottles” in a given image. This information from the pretrained model would be highly helpful in classifying each label.

III. Dataset

Yelp has provided a bundle of data which included a set of training images, test images, mappings from images to businesses for both the training images and test images via Kaggle, and labels for training businesses. The training dataset comprises of 234,842 training images and 1996 businesses, while the test dataset comprises of 237,152 images and 10,000 businesses.

Sample images in the dataset

On analyzing the training dataset we found that some labels were present extensively while there were only a few instances for the others. Around 65% of data was labelled with *has_table_service while only 35% of data had* good_for_lunch. This would mean that the model would be trained unevenly and later we shall see that our model has a high precision rate for some labels while performing poorly on the others. The dataset contains some images like the menu of the restaurant, certificates and awards received. These images actually do not contribute to any of the labels but our model cannot differentiate these outliers out leading to wrong training.

IV. System Design & Implementation

Going by the Multiple Instance Learning approach mentioned above we can broadly classify our System design into three modules.

- Input Processing Module
- CNN Training module with ImageNet.
- Adapter Module
- Multi-Class Classifier Training module.
- Wrapper Module to provide User Interface.

A. Input Processing

The training of the dataset needs to happen in an ordered way, i.e, all the images grouped under a business-id need to be trained together. Only then we will be able to generate the consolidated feature vector as mentioned previously. In order to do that we need to pre-process the dataset and arrange all the images together grouped by the business-id. This structure needs to be fed into the later modules when we start the

dataset training.

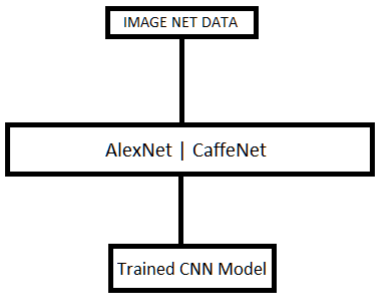
The images provided for training and testing as a part of the challenge dataset are not uniform. Some images are in landscape mode while some are in portrait. Some of the images are from facebook and instagram too. We need all the images to be uniform for the training to take place effectively. Therefore, we have pre-processed them and resized each of the image to 227 x 227.

B. CNN Training

The idea behind using a Convolutional Neural Network in our approach is to get out all the objects in the image and doing this for all the images in a particular business-id would give us an idea about the objects which are leading to the business attributes associated with the business-id.

Since the objects can be of any kind in the image, we need a neural network which has been trained on the most extensive image database available and hence AlexNet trained on the ImageNet database is a natural choice for us.

The goal of this module is therefore to train the AlexNet or the CaffeNet network with the ImageNet database and use the model later to train our muti-class classifier.



We preferred to use an already existing pre trained CNN model of AlexNet trained with the ImageNet database and available as a part of the Caffe Zoo models for our results. Other pretrained models of the VGGNet, GoogleNet, ResNet can also be used.

C. Adapter Module

This module will act as an adapter layer between the output given by the trained CNN and the input required by the Multi-Class classifier. This module can also be responsible for generating the consolidated feature vector for a given business-id.

D. Multi-Class Classifier Training

Once the CNN is trained with the ImageNet database we provide it the images from the Yelp Dataset which are grouped together with their corresponding business-id. The trained model is used to generate feature vectors for each of the image in the group and then we would calculate the mean of all these feature vectors and generate the baseline feature vector for our business-id.

This feature vector will be passed as input to train the Multi-Class classifier. We have trained a linear SVM classifier using the *onevsRest methodology. The classifier gives each of our business label a score of 1 or 0 indicating the presence of that label to the corresponding feature vector of the restaurant/business-id. By using one classifier per label, the task is reduced to multiple cases of binary classifiers predicting an output of 1 or 0 for a given label for a given business, i.e, for each label we have a binary classifier seeking to find a hyperplane dividing positive example from the negative examples. This would divide businesses having a certain label from the businesses not having that particular label.*

We used the SVM implementation of the scikit-learn library for our classifier. While training the SVM Classifier, we utilized 80% of the training businesses to train on and the remaining 20% as a validation set.

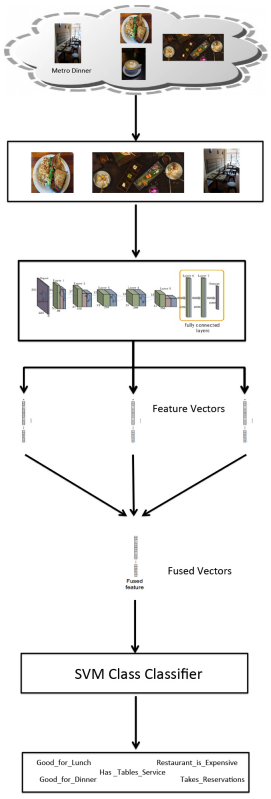
E. User Interface Module

This module would primarily be responsible to provide a user-interface for our project. The interface would take the input as a business-id and the set of images corresponding to the business-id and pass it to the underlying classification module. The classification module would run it across its components and provide the list of business attributes for the provided business-id based on its training so far back to the User Interface which would display it to the user.

We have used the Flask python framework for creating the user interface and have made our caffe model available as a backend API. The user interface can take in multiple images pertaining to a restaurant and when submitted the images are passed to our model in the backend. The model computes the labels for the restaurant which is displayed back to the user. The whole project has been deployed on a Docker container with Caffe and the required softwares installed in it.

V. FLOWCHART

The flowchart shown below is showcases our implementation plan. We take in the images of a particular business-id together and feed them to the trained CNN (AlexNet). The parameters learned would help us in understanding how we can change parts of the network in the future. For starters we hope to use the feature vectors and combine them into a single vector by taking an average. After which, they would be fed as an input to the trained Multi-Class SVM classifier. The classifier would then produce the business attributes corresponding to the business-id.



VI. Use Cases

- Attributes to restaurants will be assigned automatically and hence any user searching for a particular business attribute would get more refined results based on his choice.
- All the restaurants would be tagged with business attributes making it equally likely for each of them to appear in the search results.
- This will also help the business unit (restaurant) head to get an understanding of the current reputation of his unit in terms of well defined business attributes.

VII. Performance Evaluation

A. Experimental Setting

Training the AlexNet CNN with the ImageNet database takes a lot of time and effort. It cannot be accomplished on a local machine and needs GPU's. Therefore as suggested earlier we propose to use an already trained model of the CaffeNet on the ImageNet database available to use as a resource in Caffe. The pre-trained CaffeNet would help us in several aspects:

- ImageNet training can be skipped
- Training time of the dataset will be reduced as CaffeNet has relatively lesser layers than other models.
- CaffeNet can be trained without a GPU and hence the result can be achieved locally.

We used the pretrained AlexNet ImageNet model available in the Caffe Zoo to extract features of each image. For convenience we have named the last 3 fully connected layers in the model as FC6, FC7 and FC8 respectively. In our experiments we extracted data after each of the above mentioned layers and used the result to train the SVM described before to get results. The results of each experiment are discussed later below.

B. Performance Metrics

The performance metric given for the Yelp competition is calculated by the mean F1 score .The mean F1 score, commonly used in information retrieval, measures accuracy using the statistics precision p and recall r. Precision is the ratio of true positives (tp) to all predicted positives $(tp+fp)$. Recall is the ratio of true positives to all actual positives $(tp+fn)$. The mean F1 score is clearly depicted by the following equation

$$F1 = \frac{2pr}{p+r}$$

where

$$p = \frac{tp}{tp+fp}$$

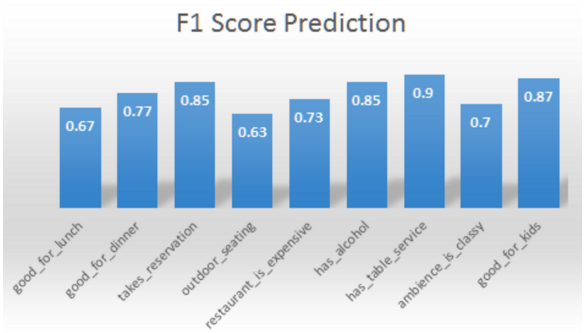
$$r = \frac{tp}{tp+fn}$$

C. Baseline Score

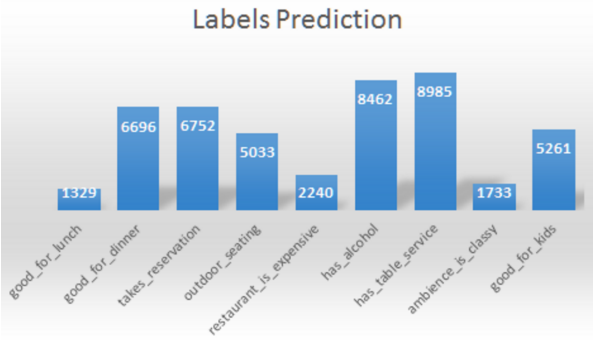
The Yelp Challenge has already give some baseline model results to refer to as comparison for our model. They claim that by randomly assigning labels to restaurants with equal probability the F1 score is roughly 0.44.

D. Transfer Learning using FC8 Data

As discussed above we first extracted data from the last fully connected layer of the pretrained network before passing it to the softmax layer. We chose to extract the FC8 layer data as it is the closest form to a human readable version of the image in terms of the ImageNet classes. Each feature vector we extracted was of 1000 dimensions and each value signified the value showcasing the probability of each ImageNet label present in our input image.



Label-Wise F1 score predicted on the 20% validation dataset

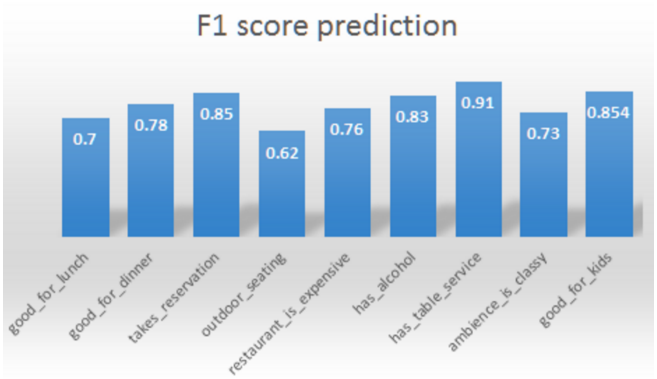


Label frequency predicted on the test dataset

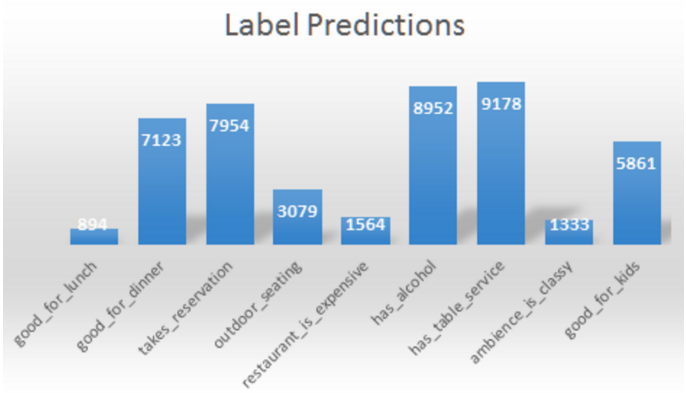
Training this data on the SVM classifier and validating it with 20% of the training dataset resulted in an F1 score of 0.793 but after submitting it in the challenge it resulted in an F1 score of 0.755 on 30% of the test dataset.

E. Transfer Learning using FC7 Data

Next we extracted data after the FC7 layer and performed the classification. The feature vector extracted here was of 4096 dimensions. We observed a slight increase in the F1 score of 0.798 for the validation dataset and a similar rise was observed on the final 30% of the test dataset.



Label-Wise F1 score predicted on the 20% validation dataset



Label frequency predicted on the test dataset

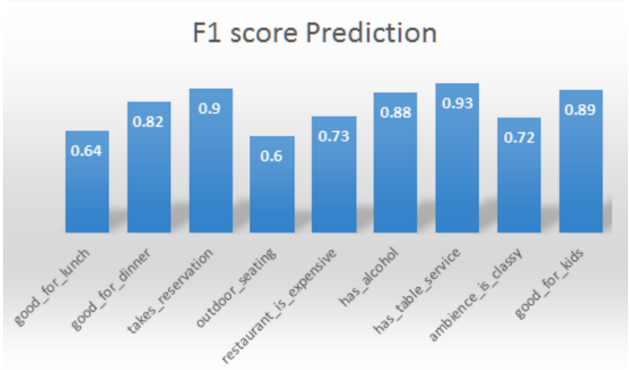
It resulted in an F1 score of 0.761. This rise can be attributed to the fact that the feature vectors given to the SVM for training have more dimensions in it and hance decision boundaries could be found easily.

F. Transfer Learning using FC6 Data

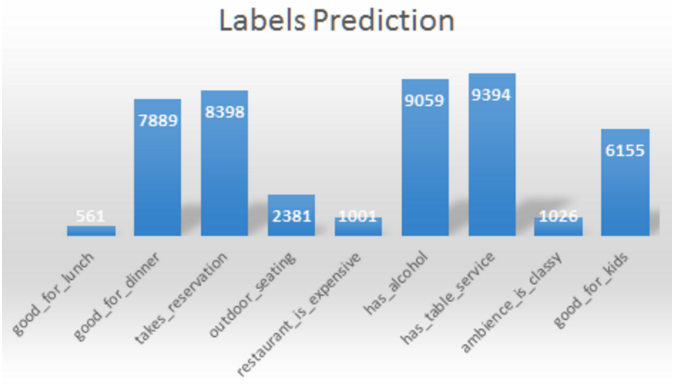
It is expected that FC6 data will lead to a lesser F1 score than the FC7 counterpart as the FC6 feature vector has lesser learned feature values when compared to FC7. We observed an F1 score of 0.782 for the validation dataset and a final F1 score of 0.752 for 30% of the test dataset.

G. Transfer Learning using FC6+FC7 Data

Comparing our results observed on the FC8 and FC7 layer data we drew a conclusion that having more dimensions in the consolidated feature vector resulted in an increase in the performance as decision boundaries could be identified relatively easily in the higher dimension.
In an attempt to further increase performance we decided to utilize the feature vectors obtained from both the FC6 and the FC7 layers and concatenate them to double the dimensionality of the consolidated feature vector. In order to consolidate the data we performed a sample wise normalization of each data first to get them to the same feature space before concatenating. The resulting consolidated feature vector can be summarised as $|FC6| + |FC7|$.



Label-Wise F1 score predicted on the 20% validation dataset



Label frequency predicted on the test dataset

Attesting our theory this approach outperformed the results of all the above approaches leading to an F1 score of 0.823 on the validation dataset and an F1 score of 0.797 on the final entire test dataset.

Observing the F1 scores of all the approaches for each individual label we have found that *has_table_service consistently performs better while labels like* good_for_lunch performed poorly. This may be because the ImageNet has many labels relating to food and table and almost every image of the dataset containing food had a table in it so naturally the preformance of *has_table_service was good. On the other hand* good_for_lunch maybe typically found by a brightly lit food image. The model may not have trained itself to recognize this or the dataset contained relatively less images signifying this.

VIII. Conclusion

For the Yelp challenge which features Multiple Instance Multiple Label learning we have found that transfer learning with an SVM classifier is a fairly good approach. In particular extracting a feature vector with high dimensions worked out as the best solution for training the SVM in this approach. As a future work we can investigate more in further fine-tuning the Transfer Learning model and also work towards different approaches for generating the consolidated feature vector to represent a particular business-id.

References

[1] Shaoqing Ren, Kaiming He, Ross Girshick and Jian Sun *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*

[2] http://cs231n.github.io/convolutional-networks/

[3] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester and Deva Ramanan, *Object Detection with Discriminatively Trained Part Based Models*

[4] Jian Yao, Sanja Fidler and Raquel Urtasun, *Describing the Scene as a Whole: Joint Object Detection, Scene Classification and Semantic Segmentation*

[5] Jason Yosinski, Jeff Clune, Yoshua Bengio and Hod Lipson, *How transferable are features in deep neural networks?*