Yelp Restaurant Photo Classification

by Anna Rumyantseva

Abstract

In December 2015, Yelp proposed "Restraurant Photo Classification" challenge on kaggle.com website. The challenge implies assigning sets of labels (e.g. "good for lunch", "has alcohol", "takes reservations" etc) to different business ids based on the corresponding users uploaded photographs of food/restaurants. The training set for the project consists of manually labelled images and corresponding business ids. To solve the challenge, transfer learning on a pre-trained Convolutional Neural Networks (CNN; VGG ConvNet model) was implemented. In particular, the pre-trained CNN was used to extract feature vectors from each image. Subsequently, a variety of supervisied learning models were applied including ensemble models. The implemented approach resulted in 0.80957 f-score (76th place on the Kaggle leaderboard).

Introduction

This capstone project aims to build a classification model that assigns multiple labels to photos of restaurants. The capstone project is based on a Kaggle Competition provided by Yelp company. 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. But Yelp's users upload an enormous amount of photos every day alongside their written reviews. This data set of restaurants photos can be turned into multiple tags using an automatic classification model. Yelp is an American multinational corporation headquartered in San Francisco, California. It develops, hosts and markets Yelp.com and the Yelp mobile app, which publish crowd-sourced reviews about local businesses.

Image Classification problem is a computer vision task that involves assigning an input image a label from a set of categories. Specifically, based on the set of images with assigned labels, one needs to build a prediction model that assigns labels to novel input images. Image classification has a wide range of practical implementations. Yelp Kaggle competition provides a great opportunity to implement image classification techniques.

Convolutional Neural Networks (hereafter CNN) revolutionized the field of computer visions by significantly outperforming previous techniques used for image recognition and classification. CNN architectures make the explicit assumption that the inputs are images. This

makes them more efficient in implementation and significantly reduce the amount of parameters in the network. Basically, CNNs arranges its neurons in three dimensions unlike regular Neural Networks. CNNs transform the original image layer by layer from the original pixel values to the final class scores. There are four main types of layers used to build CNN: Convolutional Layer, Pooling Layer, Rectified Linear Units (ReLU) layer and Fully-Connected (FC) Layer. CONV layer will compute the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume. ReLU layer will apply an elementwise activation function. POOL layer will perform a downsampling operation along the spatial dimensions (width, height), resulting in volume. FC (i.e. fully-connected) layer will compute the class scores resulting in a 1-D array.

In practice, it is not feasible to train an entire CNN from scratch since it is computationally expensive and suitable data sets for training are rare. However pre-trained CNNs can be used as a fixed feature extractor for the task of interest – transfer learning approach. Transfer learning approaching can be implemented in two ways:

1. Use outcome from Fully Connected layers of pre-trained CNNs as the image feature vectors.
2. Fine tuning parameters of pre-trained CNNs.

In this project, the transfer learning on a pre-trained CNN (using method #1) was implemented to assign multiple labels to different business ids based on corresponding images provided by Yelp users. The Yelp challenge represents the multi-class/multi-instance problem. Mutli-instance aspect of the problem comes from the fact that each business ids has multiple images associated with it. Further description of the data set and methodology is given in the next section.

Data and Methods

Yelp dataset

The data set for the project is Yelp users' uploaded images (.jpg file format) of restaurants and corresponding labels and business ids. The Data Files can be downloaded from the site: (https://www.kaggle.com/c/yelp-restaurant-photo-classification/data).

List of the Data files:

- train_photos.tgz - photos of the training set (235841 images; 6.64 GB)
- test_photos.tgz - photos of the test set (474304 images; 6.71 GB)
- train_photo_to_biz_ids.csv - maps the photo id to business id
- test_photo_to_biz_ids.csv - maps the photo id to business id
- train.csv - maps the business ids to their corresponding labels.

There are 2000 business ids in the training data set and 10000 in the test one. There are 9 labels for business IDs that can be assigned to photographs (Table 1). The top 10 of most common sets of labels in the training data set are shown in Fig. 1. Some examples of Yelp photos and corresponding sets of labels are shown in Fig. 2.

Table 1. Description of the restaurant labels.

Label #	Description
0	good_for_lunch
1	good_for_dinner
2	takes_reservations
3	outdoor_seating
4	restaurant_is_expensive
5	has_alcohol
6	has_table_service
7	ambience_is_classy
8	good_for_kids

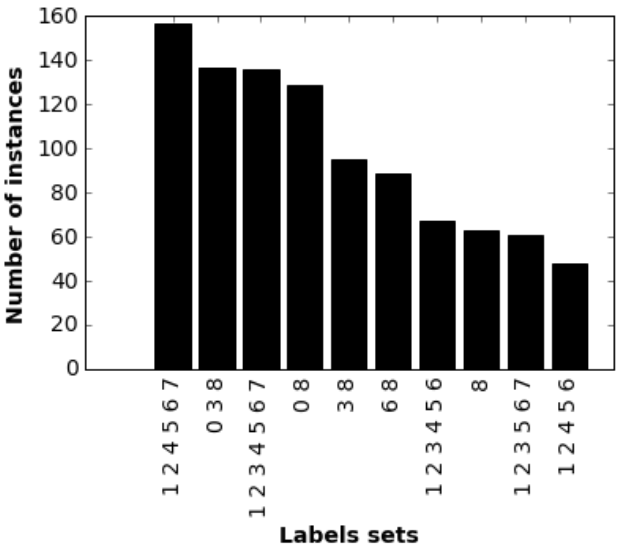

Figure 1. 10 most common sets of labels in the training data set

Good for lunch, good for dinner, good for kids

Takes reservations , restaurant is expensive, has alcohol, has table service

Good for dinner, takes reservations, has alcohol, has table service, ambience is classy

Figure 2. Examples of photographs from the training set.

Pre-trained CNN model

To extract the feature vectors (also known as CNN codes) from the Yelp images, VGG ConvNet model developed by Karen Simonyan and Andrew Zisserman was implemented. The network was trained on the ImageNet data set (1.2 million images with 1000 categories) and became a runner-up for the Large Scale Visual Recognition Challenge 2014 (ILSVRC2014). The winner of the challenge was GoogLeNet. But, it was shown that VGG ConvNet features outperform those of GoogLeNet in multiple transfer learning tasks.

The images were processed through the pre-trained model using MatConvNet Matlab toolbox. MatConvNet is a MATLAB toolbox implementing CNNs for computer vision applications with many pre-trained CNNs for image classification are available in the toolbox. Fast VGG architecture (imagenet-vgg-f.mat 216 Mb; scheme of CNN is available here http://www.vlfeat.org/matconvnet/models/imagenet-vgg-f.svg) was downloaded for the feature extraction. Average image (available in net.meta.normalization.averageImage) was subtracted from all the images before running them through the model. The features vectors were taken as an outcome of the fully connected layer #6, #7, #8 and the probability layer. Hereafter the names of the layers are abbreviated as follows fc6, fc7, fc8, prob. The length of CNN vectors for the layers are 50726, 4096, 4096 and 1000 respectively.

Classification

Based on the obtained CNN codes the following steps were implemented for assigning labels to business ids:

1. Calculate mean CNN code vectors (for fc6, fc7, fc8 and prob layers) for each business id in train and test data sets ignoring duplicates. These mean CNN codes were used as feature vectors in subsequent machine learning process. This step deals with the **multi-instance aspect** of this problem.
2. To deal with **the multi-label aspect**, One-vs-the-rest (OvR) multiclass/multilabel strategy was applied (sklearn.multiclass.OneVsRestClassifier function in python). One-vs-the-rest strategy consists in fitting one classifier per class. For each classifier, the class is fitted against all the other classes. OneVsRestClassifier was ran on top of the following supervised learning models:
    - Logistic Regression (LR)
    - Support Vector Classifier (SVC)
    - Decision Tree Classifier (DT)
    - K Nearest Neighbour (KNN)

- Gaussian Naive Bayes (GNB)
- Random Forest (RF)
- Gradient Boosting (GB)

Models were fit using different combinations of feature vectors (for some of the classifiers) and GridSearchCV function, which searches through a grid of parameters for each model. Since the feature vectors are quite long, especially when concatenating mean CNN codes from different layers, PCA decomposition was tested.

3. Make a prediction on feature vectors for unlabelled business ids in the test set. The voting technique was implemented in order to check if the combined outcome from the different model can improve the prediction.

The models were evaluated using **f-score** metric, as suggested by the Kaggle competition. The metric considers both the precision p and the recall r of the test to compute the score:

$$f = 2 * \frac{p * r}{p + r}.$$

Results

Power of CNNs

The Fig 3. represents a word cloud of image scores from the trained VGG-Convnet model derived from Yelp images. The word cloud shows that the absolute majority of images scores are "food-related", with "plate" score being the most common. It shows that the CNN gives sensible outcome for the input images.

Figure 3. Images scores from VGG Convnet derived from Yelp images.

Logistic Regression (LR)

First of all, cross validation (k-fold with 5 folds) on Logistic Regression was used to try the performance of a different combination of features and PCA decomposition. Cross validation showed that the best performance was achieved using PCA decomposition on 100 components (Fig. 4). The best performing combination of features for Logistic regression was PCA decomposition on concatenated mean CNN codes from layers fc7 and fc8 (fc7_fc8). After selecting the best performing feature, GridSearch of the best parameters (C (inverse of regularization strength) and penalty type) was implemented. The achieved f-score on cross validation was **0.82818**. **For other classifiers only features PCA decomposed to 100 components were considered since they show the highest score and decrease the time of running classifiers.**

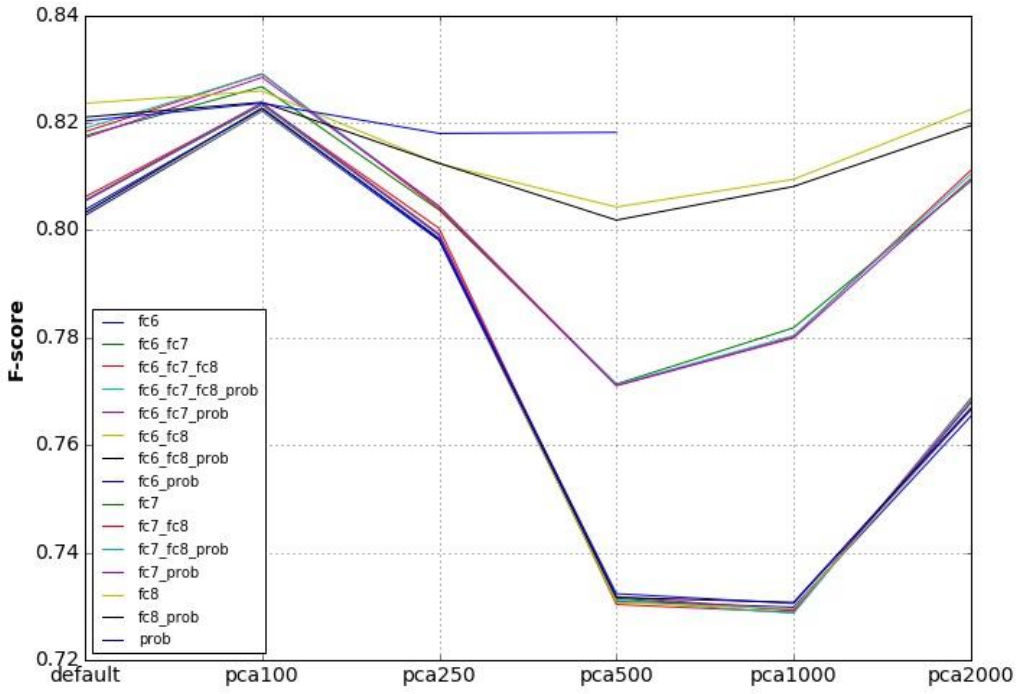

Figure 4. Mean F-score obtained from k-fold cross validation using different combinations of features and PCA decomposition.

Support Vector Classifier (SVC)

Running SVC classifier on all the feature combination takes a long time. Therefore classification using SVC was implemented only to fc7_fc8, the best performing feature combination for the Logistic Regression. The Grid Search for parameters showed that the best

score is achieved by using rbf kernel with kernel coefficient gamma = 0.0001 and penalty parameter C =1. The achieved f-score on cross validation was **0.83254**.

Decision Tree Classifier (DT)

The Decision Tree Classifier was applied to all feature combinations. Similar to Logistic Regression, the best performing feature combination was fc7_fc8. Grid search for parameters was also implemented to find the best combination for maximum depth of the tree and the minimum number of samples required to split a node. The achieved f-score on cross validation was **0.77096**.

K Nearest Neighbourgh (KNN)

For KNN classifier the best performing feature was the outcome from fc8 layer when 20 was used as the number of neighbours in the classifier parameter set. The achieved f-score on cross validation was **0.80608**.

Gaussian Naïve Bayes (GNB)

The Gaussian Naïve Bayes classifier was also tested for all feature combinations. The best performing feature was the outcome from the fc8 layer. However, the performance of the classifier was significantly lower compared to other ones. The achieved f-score on cross validation was **0.694705**.

The outcome from cross-validation/feature selection is summarized in Table 2. Random forest and Gradient Boosting classifiers showed the very low f-score on cross validation: 0.18 and 0.16 respectively. Therefore, they were not considered in subsequent steps.

Table 2. Performance of classifiers on cross validation

Classifier	**Mean f-score from CV***	**Selected features**
Logistic Regression	**0.82818**	fc7_fc8
Support Vector Classifier	**0.83254**	fc7_fc8
Decision Tree	**0.77096**	fc7_fc8
K Nearest Neighbour	**0.80608**	fc8
Gaussian Naïve Bayes	**0.694705**	fc8

* CV stands for Cross Validation

Voting

Several classifiers have been tested and several of them showed fairly good performance. The outcome from different classifiers can be ensemble together in order to improve the accuracy of models. Model can be combined in several ways:

1. Majority voting using predictions from the classifiers. Simply, if 3 out of 5 classifiers predicted 1, the final prediction is 1. One can also assign weights for voting. Such as the better performing classifiers have more weight in vote compared to the rest. Different combinations of weights have been tested.
2. Another approach implies training a classifier on top of the predictions from the classifiers. In this project, I trained a simple Logistic Regression on top of the predictions from LR, SVC, DT, KNN, GNB classifiers obtained from the train set.

Both of the methods have been implied. The majority voting and weighted voting appeared to decrease the accuracy of the model.

Submission to Kaggle

Table 3 shows results of the submissions of predicted test labels to the Kaggle website. In total, I have submitted 5 model: Logistic Regression Classifier, Support Vector Classifier with rbf kernel, Ensemble learning based on majority vote, Ensemble learning based on weighted vote and Ensemble learning stacked with Logistic Regression. The best score was achieved for a simple models: Support Vector Classifier. The ensemble learning method has not improved the final result on the Kaggle leaderboard for the competition. Compared to majority vote and weighted vote, ensemble learning with stacked logistic regression performed significantly better.

Table 3. Kaggle results

Model	**f-score**	**Place on the Leaderboard**
Logistic Regression Classifier	0.80184	95
Support Vector Classifier (rbf kernel)	**0.80957**	**76**
Ensemble Learning based on majority vote	0.74898	146
Ensemble Learning based on weighted vote	0.62099	280
Ensemble learning stacked with Logistic Regression	0.80538	86

Conclusions

This Capstone project aimed to build a solution for "Yelp Restaurant Photo classification" Kaggle Challenge. The challenge represented multi-instance, the multiclass machine learning problem. Classification of photographs was conducted by implementing transfer learning approach on the pre-trained CNN model. The outcome of the project showed that:

- Classification of images using the outcome just from one CNN model results in relatively good f-score (~0.80).
- The models show higher accuracy when PCA decomposition on 100 components is applied on feature vectors for images. It also significantly decreases the time of training the models.
- In this project, simple model (SVC classifier) outperformed the ensemble models. f-score on the test set for SVC classifier was 0.80957 that brought me to the 76th place on the Kaggle leaderboard for the competition.
- Further improvements in the model performance can be achieved by advanced feature engineering. The winner of the competition, Dmitrii Tsybulevskii, also implemented the transfer learning approach for tackling the problem (http://blog.kaggle.com/2016/04/28/yelp-restaurant-photo-classification-winners-interview-1st-place-dmitrii-tsybulevskii/). However, he used the outcome from several pre-trained CNN models as feature vectors for classification. f-score achieved by Dmitrii was 0.83177.

Acknowledgements

I would like to thank you to my mentor Alex Chao for assisting me with the project. His encouragement, tips and suggestions helped me enormously. Also, I would like to express my gratitude to the whole SpringBoard team for putting together this amazing course!

Appendix 1.

I have tested the prediction model on some of my photographs from restaurants/bars/cafes ☺

Photo 1. Beers in a bar in New Orleans: good for lunch, has alcohol

Photo 2. Homemade fancy dinner with a friend: good for dinner, takes reservations, restaurant is expensive, has alcohol, has table service

Photo 3. Having coffee in Cafe du Monde in New Orleans: good for lunch, good for kids

Photo 4. Dinner in Japan: good for lunch, has a table service, good for kids

Photo 7. Beer in a street café in Spain: has alcohol, has table services

Photo 8. Street food in Japan: has table services, good for kids, good for lunch

Appendix 2.

The GitHub repository for the Capstone project contains several ipython notebooks:

- *Yelp_images_machine_learning.ipynb* is the notebook with codes for training classifiers and there comparison
- *Yelp_images_data_wrangling.ipynb* is the notebook that prepares data frames for machine learning. In particular, the code loads Matlab files with CNN codes and calculates feature vectors for each business.
- *Yelp_images_cloud_words.ipynb* is the notebook in which the word cloud (Fig. 3) is created