Restaurant Photo Classification using Deep Convolutional Neural Networks

Chennakesava Manikanta Ijjada
Department of Computer Sciences
University of Florida
Gainesville, Florida, USA
chennakesava@ufl.edu

Giridhar Ganapavarapu
Department of Computer Sciences
University of Florida
Gainesville, Florida, USA
gganapavarapu@ufl.edu

Praneeth Rajput
Department of Computer Sciences
University of Florida
Gainesville, Florida, USA
prajput@ufl.edu

***Abstract* - We aspire to address the image classification problem specifically in the context of online restaurant review portals which witness unprecedented unlabeled image uploads. This massive dataset presents an untapped potential to enrich the user interaction with experiences such as tabbed photo browsing. We propose to train a deep convolutional neural network to classify images into a predetermined set of classes as part of the 'Yelp Restaurant Photo Classification' [7] challenge at Kaggle.com. In this project we have tackled a multi-label classification problem by classifying the images into 9 concepts. To further extend our implementation by reusing above mentioned classification to power the search results by leveraging Elasticsearch.**

Keywords—CNN, CaffeNet , Caffe, REST, Jersey, Docker, Elasticsearch

1. INTRODUCTION

Image classification analyzes the properties of images and segregates the data set into useful categories. The classification algorithms typically involve the training and testing phases, the former involves creation of unique training classes based on the training data while the latter employs test data to classify image features. The training classes are independent, discriminatory and reliable which imperatively ensures that there is a significant difference between different images, the classification is reliable in the sense that images in each class share a common definitive description [10]. The current problem statement that we intend to address is photo classification at Yelp, the extensive user base is a producer of significantly important unlabeled data sets. We hope to design and develop a model to classify such images into a set of 9 descriptors which leads to a potential game changer in the way the images are consumed by the community. Yelp is home to millions of reviewers who upload images which are untagged. A classifier model trained on this dataset can eventually be ported into a RESTful interface and the descriptor set can be a potential search criterion on such portals. The training and test data needed for this exercise is available as part of the Kaggle competition [7].

2. REALTED WORK

2.1 ImageNet Classification

ImageNet classification which trains a deep convolutional neural network for the classification of high resolution images achieves a significant breakthrough in the field of image classification [1]. The paper claims to have achieved a top-5 error rate of 17% on test data, top-5 error rate corresponds to instances where the image was not classified amongst the top

5 probable labels, which is the basis for related work done by Yelp in Image Classification. Yelp developed a photo classifier based on intrinsic and crowd sourced data sets to realize a classification system to power the tabbed photo browsing experience on the portal. The authors of this paper intend to leverage this learning to deploy a classification system based on the below explained architecture model.

2.2 Multi-Instance Multi-label Learning

The paper on Multi-Instance Multi-Label learning with Application to Scene Classification by Zhou et al. [11] formalizes MIML wherein every training example can not only have multiple instances but also multiple labels associated with it. It also proposes the MIMLBOOST and MIMLSVM algorithms in context of scene detection and are shown to provide good performance. As opposed to the traditional supervised learning techniques where the training example is represented by an instance and is only associated with a single label, MIML learns a function to tackle training examples which can be associated with multiple labels and are described by multiple instances. Attached below is the pictorial representation of the four learning frameworks.

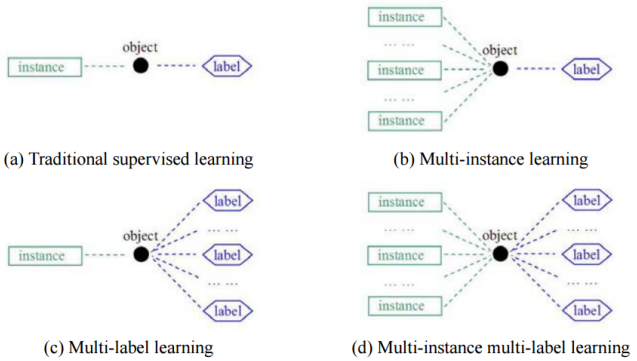

Figure 1: Learning Frameworks [11]

3. ARCHITECTURE

A brief description of the components employed in the solution are presented below:

3.1 Convolutional Neural Network:

Neural Networks can be layers of neurons receiving an input which is passed down to hidden layers for transformation with the output being realized in the Output Layer. Convolutional Neural Network [3] (CNN) is similar to neural network with learnable weights and biases. Each neuron performs a dot product transforming the image raw pixels in our case to a differential score. Convolutional Neural Networks are best assumptions in case of images and allows to encode properties into the architecture.

A ConvNet is generally made up of various layers where each will transform an input 3-D data into an output 3-D data with a function that may or may not have parameters. There are three main layers to build Convolutional Neural Networks namely Convolutional Layer, Pooling Layer and Fully Connected Layer. Convolutional Layer is core building block with its output holding the neurons in 3-D volume. Pooling layer is commonly put in between convolutional layers in a ConvNet architecture. Its function is to progressively reduce the spatial size in order to reduce computation and parameters in the network. Fully Connected layer consists of neurons having full connections to activations in the previous layer. These are calculated by matrix multiplication which are followed by bias offset. There are various architectures in the field of Convolutional networks such as LeNet, AlexNet, ZFNet, GoogleNet and VGGNet.

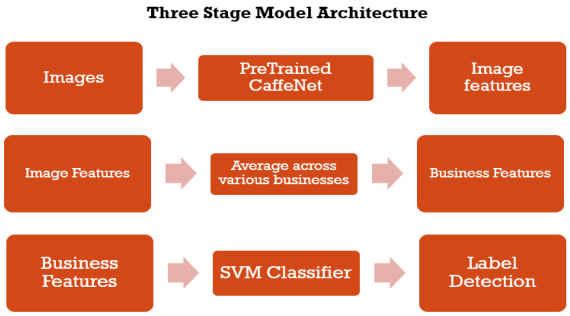

Figure 2: Architecture Model

3.2 Deep Learning Framework

Among various deep learning frameworks, we are using Caffe which is developed by Leaning Center and Berkeley Vision. It's an expressive architecture which encourages innovation and application where one can manage the models and optimization without hard coding the configurations. It provides the facility to switch between GPU and CPU during training process using a single flag. Extensible nature of Caffe enables active development and encourages contributors to make significant changes to the framework. Caffe will be a perfect framework for our project as it has the ability to process more than 60M images per day with very high learning and inference rates and is currently fastest ConvNet implementation.

3.3 Transfer Learning

The main idea of transfer learning from an architecture like CaffeNet was an effective strategy. The Main concept of transfer learning is to use a network that was trained for ImageNet challenge in 2012/2013 for other image classification challenges and secure a state of the art level accuracy. Various studies concluded that fine tuning the weights of a pre-trained network could help us in attaining good performance. Transfer learning can be explored in two different scenarios, First scenario helps in adding different layers to existing pre-trained layers and achieve our custom output. In second scenario, pre-trained layers can have a low learning rate and get pre-trained weights but helps in attaining fine-times for new features with training.

3.4 Application Design

We intend to build a RESTful service using Jersey framework which aids in classifying the photos into relevant descriptor sets. RESTful applications essentially depend upon HTTP protocol to handle CRUD operations and are light weight frameworks alternative to SOAP.

While the challenge is limited to tackling a multi-label classification problem of classifying the images into 9 concepts, we intend to provide a holistic design by incorporating these results as part of a search solution. Elasticsearch is a search server based on Lucene which provides a distributed real-time search engine with high availability and multi tenancy. It provides powerful full-text search capabilities that are API driven. [9]

4. SYSTEM IMPLEMENTATION

The challenge is to predict the attribute levels for restaurants utilizing the community uploaded images. We are given set of photos that belong to a business and are required to predict the below mentioned nine attributes:

	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

In order to develop a classifier that will put given photos into specified attributes, we have to collect photos with known labels. For this purpose we will be using the dataset provided by Yelp [2] as part of the competition. We have utilized MIML with transfer learning along with a Linear SVM classifier to build this model and propose the following three stage architecture:

From below shown wrapper model we can combine two classifiers through their weighted averages.

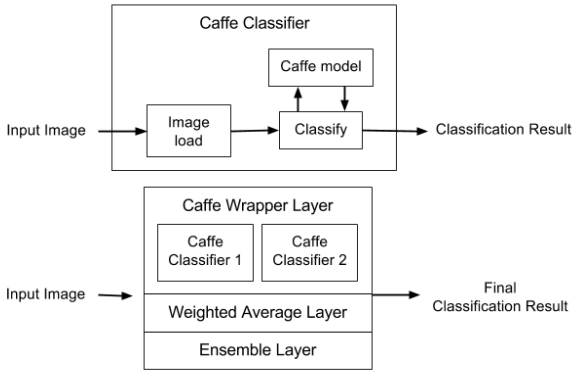

Figure 3: Classifier Model [7]

4.1 Data Set:

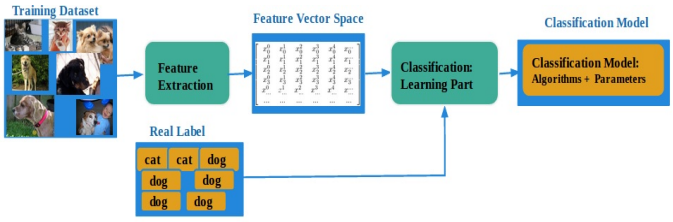

Figure 4: Training the Model

The training and test data set comprise of images uploaded by users. Business attributes are to be predicted using the features predicted from photos uploaded by business users. The uploaded photos possibly can have content which is not at all related to the label. It is possible that same photo is uploaded to the same business more than once. And the same photo can be uploaded to different branches in case of chain business. These issues make this problem a perfect application for ConvNets. Each photo is associated to a business id. We obtain all the features from business photos and then identify the business model.

Following are the details of the data set for the Yelp coding challenge in Kaggle:

- train_photos.tgz: is of size 6.54 GB has images of training data set.
- test_photos.tgz: is of size 6.61 GB and has images of test set.
- train_photo_to_biz_ids.csv: is of size 1.12 MB and has the information to map business id and image id of training data set.
- test_photo_to_biz_ids.csv: is of size 4.79 MB and has the information to map business id and image id of test data.
- train.csv: is of size 7.12 kb and has the information about the labels for each business id.
- sample_submission.csv: is of size 38.98 kb and is format for business id and predicted business feature labels.

4.2 Implementation:

The proposed algorithm is to utilize transfer learning paired with a linear SVM. The images have been fed through a CaffeNet model pre trained on the ImageNet dataset. This procedure of feeding images through a networks pre trained on ImageNet is done through up to FC-6 and FC-7 layers so as to obtain a vector for each image. The following three stage procedure has been implemented to realize the classification of images.

4.2.1 Feature Extraction

The images in both Training and Test datasets have been preprocessed and resized to 227 * 227. The training image set has been fed into a pre trained CaffeNet model with a batch size of 700 for the feature extraction stage. The extracted features are stored as part of an h5 file for later processing having an internal structure of filenames and features. This stage of processing has an approximate 336 iterations for the Yelp dataset. Above mentioned methodology is applied to the test dataset as well.

4.2.2 Business Feature Vector Calculation

The problem statement is considered as Multi Instance learning problem where each business is described by multiple instances. A 4096 dimension feature vector has been obtained for every image and a mean feature vector has been computed for each business. Utilizing the Image Id – Feature Vector data computed in stage 1 along with Business Id – Image Id data provided as part of the dataset, the Business Id – Feature Vector data is consolidated and the above mentioned mean feature vector for each business is computed. The feature vectors are computed as an array of 'business', 'label', 'feature vector'. The Business Feature Vector is calculated in the similar manner for the Test dataset as an array of 'business' and 'feature vector'.

4.2.3 Linear SVM Classifier

In the final stage, a linear SVM is trained to predict the 9 concepts for each business where each label is attributed a score of 1 or 0 based on its presence or absence in the image. A linear mapping is trained against the business feature vectors which computes the score for each label.

	business	label	feature vector
0	1000.0	(1, 2, 3, 4, 5, 6, 7)	[0.1997723, 0.43287033, 0.2273244, 0.35517699,...
1	1001.0	(0, 1, 6, 8)	[0.0, 0.58892852, 0.53906053, 0.17221896, 0.01...
2	100.0	(1, 2, 4, 5, 6, 7)	[0.1115494, 0.034822516, 0.12025651, 0.5201101...
3	1006.0	(1, 2, 4, 5, 6)	[0.078059554, 0.054452702, 0.056381796, 0.6942...
4	1010.0	(0, 6, 8)	[0.3965643, 0.27963245, 0.0, 0.17205055, 0.361...

Figure 5: Sample Business Vectors for Training Data

	business	feature vector
0	003sg	[0.1930487, 0.25836372, 0.19439946, 0.46233258...
1	00er5	[0.19397442, 0.2554754, 0.18416312, 0.33579403...
2	00kad	[0.12131335, 0.1265531, 0.076526575, 0.3834464...
3	00mc6	[0.28429005, 0.11110999, 0.47850242, 0.4494442...
4	00q7x	[0.23811702, 0.33040076, 0.25543568, 0.3258069...

Figure 6: Sample Business Vectors for Test Data

4.2.3 SVM Classifier

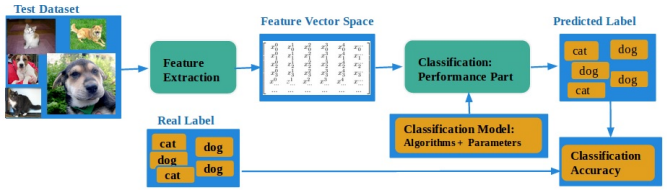

Figure 7: Testing the model

The above figure describes the process of prediction and performance evaluation. Features of images are classified by the designed algorithm and predicted with possibly few failures. Predicted business features are computed and compared to true business features in Classification Accuracy part. The question appears that how to distinguish the features of different business.

A restaurant is considered to be good for lunch or dinner based on the type of food served like pizza, sandwich, burger etc. Outdoor ambience like roof, greenery, lighting helps to identify restaurants with outdoor seating. Labels like lobster, posh, clean would help us know whether a restaurant is expensive. The color of the drinks in the glasses including the type of the glasses would help us identify restaurants with alcohol. The arrangement and structure of tables in the dining area will be key to find out restaurants with table service. Type of food and other features like alcohol availability would be helpful to identify restaurants that are good for kids. It might be difficult to classify a business as classy but some of the expensive restaurants would come under classy category.

4.3 Search API Layer Implementation

As described in the abstract of this paper we have made an effort to provide a holistic design solution where once an image is classified into one or many of the 9 concepts, this information can be leveraged to power the search or serve as a metadata store for the restaurant owners. We intend to provide a HTTP interface to our final model deployment where post classification an asynchronous event is fired to tag this information into an instance of Elasticsearch. We also intend to provide an API driven solution to query these results from Elasticsearch. This final block of implementation will serve as the demo-able layer of our project. The request/response specs for the Restful APIs are attached below in the Appendix section.

4.4 Environment & Configuration

We have trained and implemented the initial dataset on the following hardware configurations:

Configuration1	Intel i3 Dual Core CPU
Configuration2	Intel i7 Quad Core CPU

We intend to extend our final deployment to an Amazon EC2 GPU machine and port the implementation along with the Restful API leveraging Elasticsearch.

5. PERFORMANCE EVALUATION

By utilizing the Transfer Learning approach in tandem with a Linear SVM classifier we were able to realize a classification system for the Yelp dataset. In the below sections an elaborate evaluation strategy along with the results are available.

5.1 Evaluation Properties:

The correctness of the models can be evaluated from precision, accuracy and recall. A standard metric for a classification problem is accuracy and it will allow us to compare prediction accuracy of our model with other models and help us in fine tuning the parameters of our models as we move forward. In current Yelp challenge we are primarily interested in predicting the true positive attributes. For example, when a query asks for a specific attribute, we would like to return all the images with the given attribute, hence the true positives. Accuracy scores depends on how much correctly images are predicted but this is not main basis for our system output.

Before we go forward on precision and recall, let's discuss important terminologies using an attribute alpha.

True Positive (TP)	Group of annotators that predicts the alpha attribute and alpha is one of the CNN top-n predictions.
False Positive (FP)	Group which predicts that the alpha attribute is not in the image and the CNN predicts the alpha attribute.
False Negative (FN)	Group of annotators predicts the alpha attribute and the CNN predicts the unknown noise class.
True Negative (TN)	Group of annotators predicts the alpha attribute is not in the image and the CNN predicts unknown noise class.

The measure of T P / (T P +F P) for a given search attribute will represent how many of the returned images are relevant to searched attribute and this specifies the precision. An attribute with lower precision means that filtering is not happening up to the mark and human intervention is needed to filter the images predicted by the CNN model for a given attribute. So

precision will show how well given model is working. In contract, Recall is a measure which signifies the how many images for a given attribute are identified by CNN and it is represented as TP / (TP + FN). Lower recall values means that we are unable to find the images representing a given attribute and in that case, CNNs are not good at finding the attributes and human intervention is need. However lower recall value is not as detrimental as a lower precision and a good model need to maximize precision keeping recall and accuracy at specified bounds.

5.2 Performance Metrics:

As part of this project we are considering the F1 score as a metric for performance evaluation:

```
F1 score:  0.816065192084
Individual Class F1 score:  [ 0.71014493  0.79396985  0.87254902  0.66341463  0.69811321  0.84773663
  0.93430657  0.80357143  0.86919831]


F1 score:  0.795467751307
Individual Class F1 score:  [ 0.651341    0.78481013  0.82878412  0.5914787   0.77272727  0.84040404
  0.91992551  0.75949367  0.86464646]
```

Figure 8: F1 score for FC-7 vs FC-6 layer

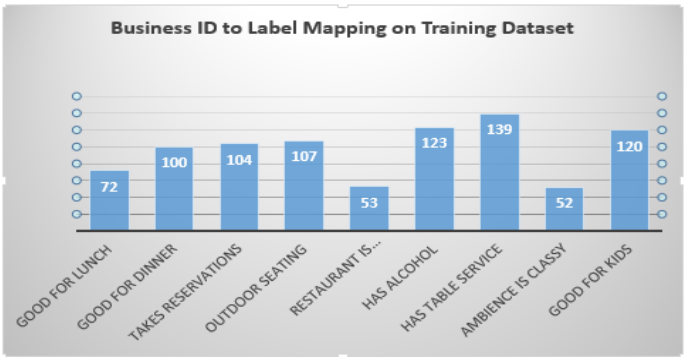

Figure 9: Business ID - Label Mapping for Training Set

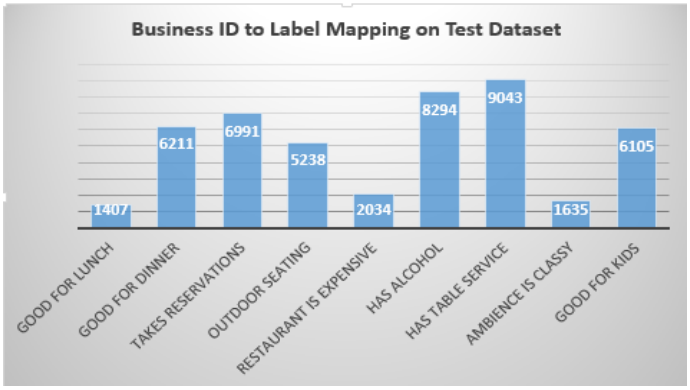

Figure 10: Business Id - Label Mapping for Test Set

Figure 11: Tagged Good for Lunch

Figure 12: Tagged Good for Dinner

Figure13: Tagged as Taking Reservations

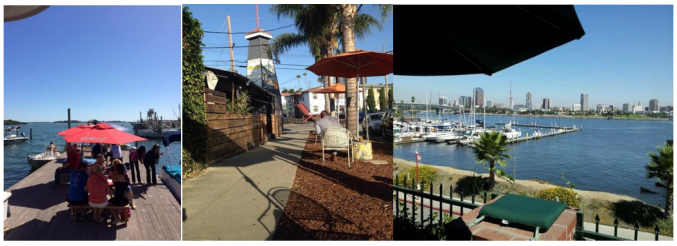

Figure 14: Tagged as Outdoor Seating

Figure 15: Tagged as Expensive

Figure 16: Tagged as having alcohol

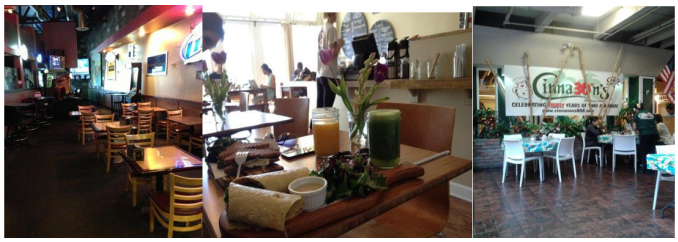

Figure 17: Tagged as having Table Service

Figure 18: Tagged as Classy Ambience

Figure 19: Tagged as Good for Kids

6. CONCLUSION

By utilizing the above mentioned design a model has been developed that tags restaurants with multiple labels based on a dataset of community uploaded images and a F1 score of 0.76 has been realized. Also, an attempt has been made to design a holistic solve for the problem by reusing the tagged attributes in Elasticsearch. Restful APIs have been developed to automatically tag images and also search restaurants based on the 9 concepts mentioned in the challenge.

We are confident that further improvement can be made with regard to performance and accuracy by running the feature extraction on FC-8 layer and perform the pre-training by doubling the feature maps in convolutional layers or by increasing the number of middle layers

7. REFERENCES

[1] *ImageNet Classification with Deep Convolutional Neural Networks:* http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf

[2] *Yelp Restaurant Photo Classification:*
https://www.kaggle.com/c/yelp-restaurant-photo-classification/data

[3] *Convolutional Neural Networks:*
http://cs231n.github.io/convolutional-networks/

[4] *How We Use Deep Learning to Classify Business Photos at Yelp:*
http://engineeringblog.yelp.com/2015/10/how-we-use-deep-learning-to-classify-business-photos-at-yelp.html

[5] *Neural Networks and Deep Learning:*
http://neuralnetworksanddeeplearning.com/

[6] *Gradient based learning applied to document recognition:*
http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf

[7] *Yelp Restaurant Photo Classification*
https://www.kaggle.com/c/yelp-restaurant-photo-classification

[8] *Image Classification: Analysis*
http://homepages.inf.ed.ac.uk/rbf/HIPR2/classify.htm

[9] *Wikipedia Elasticsearch*
https://en.wikipedia.org/wiki/Elasticsearch

[10] *Affective Image Classification*
http://www.imageemotion.org/machajdik_hanbury_affective_image_classification.pdf

[11] *Multi-Instance Multi-Label learning*
http://papers.nips.cc/paper/3047-multi-instance-multi-label-learning-with-application-to-scene-classification.pdf

[12] Deep Detect Image Tagging
http://www.deepdetect.com/tutorials/es-image-classifier/

8. APPENDIX

8.1 Project Management

The current status constitutes about 90% of the project work we had initially envisaged. The implementation of a Restful API layer is an addition to our mid –term proposal document as we felt the need to provide a holistic design solution and also provide an interface to demonstrate the project. As a team with regard to completion of project, we are yet to try out other alternatives to better the F1 score. We intend to complete that by the final demonstration of our project and include the findings as part of the final report.

All the individual tasks have been created in Asana to provide an overview of team and individual contribution. We also intend to provide a snapshot of team and individual deliverables during the final demo of the product which will detail out the major milestones and timelines of our project and also provide an overview of every team member's contribution. We are extremely satisfied with the effort we have put in with regard to this project and are thrilled with the opportunity provided as part of the course to work on such a deliverable.

8.2 REST API Request/Response Specs

[Yet to include]