**ECE324 Course Project Proposal**

**Grant Lau, Vince Tran, Hshmat Sahak**

**Problem Statement**

This project aims to predict Yelp and Google Maps restaurant ratings using photos from customer reviews. We want to determine how well a model can predict the quality of a restaurant using only image data and compare the accuracy of image-based predictions to the accuracy of predictions generated from metadata and/or sentiment analysis of text reviews. This is interesting for multiple reasons. Firstly, many review sites allow users to post images without an associated rating. Generating a rating makes an image-only review considerably more useful for the purposes of determining the relative quality of different businesses. Secondly, the project allows social media platforms with image data to create travel guide or business recommendation features for users.

**Data Gathering Plan**

Data will be gathered using the Yelp and Google Maps APIs as well as the Scrapy web-scraping framework. Both the Yelp and Google Maps APIs function similarly, allowing users to retrieve a list of relevant businesses based on a search query and a location. Repeated calls to the two APIs will be made using combinations of a search query and a location. The list of possible search queries will look like [‘French restaurants’, ‘Chinese restaurants’, ‘Indian restaurants’, etc.] and the list of possible locations will look like [‘Toronto’, ‘Montreal’, ‘Vancouver’, etc.]. The Google Maps API provides a business ID, rating, and up to 10 photographs for each business. The Yelp API provides a place ID, rating and three photographs per business, but an additional 9 photographs can be acquired through web scraping. The business ID and place ID will ensure data on a business is not collected twice on the Yelp and Google Maps APIs respectively. We aim to collect approximately 10,000 photographs, each with an associated restaurant rating. Data augmentation techniques such as cropping, flipping, noise injection may be considered to expand the dataset. All images will be resized to 256x256 before being fed into our CNN.

Below are the API specific procedures to retrieve relevant restaurant photo and rating data.

Yelp API Procedure

1. GET https://api.yelp.com/v3/businesses/search to search for *business IDs*
2. GET [https://api.yelp.com/v3/businesses/{id](https://api.yelp.com/v3/businesses/%7Bid)} to get restaurant *name* and *rating*
3. Scrape <https://www.yelp.ca/biz_photos/scrollable_photos/>{id} for “top” 12 photos

Google Maps API Procedure

1. GET [https://maps.googleapis.com/maps/api/place/textsearch/*output*?*parameters*](https://maps.googleapis.com/maps/api/place/textsearch/output?parameters)to search for *place IDs, names* and *ratings*.
2. GET [https://maps.googleapis.com/maps/api/place/details/*output*?*parameters*](https://maps.googleapis.com/maps/api/place/details/output?parameters)to get 10 photographs of the business.

As a backup plan or to further expand the dataset, Yelp provides a Photos dataset [1] which includes 200,000 photos with associated business IDs. The Yelp API can be used to get the rating for each of the businesses.

**Prior Work:** For our prior work, we value papers that use similar dataset to answer similar questions.

1. **Predicting Restaurants’ Rating And Popularity Based On Yelp Dataset [2]**

The goal of this paper was to analyze several ML models to determine which performs best at predicting restaurants’ rating and popularity based on data from the Yelp Dataset. The paper is a good pick because it is written by students as part of a project, so it is a realistic example of the kind of analysis we could do. More importantly, they use the Yelp Database and investigate many ML models. Unlike the ensemble approach we discuss next, the output of all rating-based models is a number from 1-5, discretized to increments of 0.5 This is exactly what we want for our model’s output, so it is worthwhile to investigate the paper’s approach.

The data of the paper comes from the Yelp Dataset Challenge. They focus on restaurants in Toronto- 5000 is used for training and 1750 for testing. Another 5700 images near Toronto are included in testing to assess generalization. Unlike our model which inputs images, they use restaurant features like available services, price level, locations, opening hours, etc. In total, 42 features are used. While input is different from our model, it is still non-textual information, so gives us an idea of how successful region-based instead of user-based analysis is (note: we assume images of food would be user-independent with respect to model outputs).

Before assessing performance for various ML models, the authors conduct a forward feature selection using MATLAB “sequentialfs” function. This is done to avoid overfitting and only include a number of features that keeps loss low enough.

The first model is linear regression, with selected features from the feature selection above. To calculate prediction error, they use a simple discretization rule. A continuous variable is predicted by the regression, and then rounded to the nearest 0.5. This way, results are comparable with other methods we now describe. The second model used multinomial logistic regression. This tries to predict the probability of dependent variable falling into each category and selects the class with highest probability. The third model is Naive Bayes model. Here, the class with the highest conditional probability of output given restaurant features is selected. The authors choose a multinomial prior for the feature vector, since all variables are discrete or are discretized. The prior and conditional probabilities are both attainable from the dataset. The final model is a 3-layer neural network with hidden layer size 100, and sigmoid activation. MATLAB is used for training all models.

The result of experiments is that linear and logistic regression perform slightly better than Naive Bayes and Neural Network. Their best model is logistic regression and has accuracy of 32%. According to the authors, they don’t have enough input features for a decent prediction. They claim they can benefit from more features that capture factors affecting ratings and also more training samples. Furthermore, model performs better for restaurants in Toronto than near Toronto. However, they note that the forward feature selection is successful in solving overfitting. The problem was thus identified as data being very local and only representative of Toronto.

This paper highlights the importance of using representative data. According to [3], the most important factor for the success of a restaurant is the food. By using food images on Yelp, we will have both data that is much more in quantity and more representative of factors affecting rating. If we can use a CNN or other architecture to extract features from food images or restaurants and use this instead of metadata (as in this paper), our project can be seen as an improvement to this paper; especially if we use their techniques and come up with better results.

1. **A Deep Learning Ensemble approach to the Yelp Restaurant Classification Challenge [4]**

This paper attempts to train a deep learning model to label restaurants with multiple categories based on user submitted photos of the restaurant. Specifically, they use the Yelp Restaurant Classification Challenge (YRCC) data to train several CNNs and select the one with the highest F1-score in the YRCC. The final model is a multi-model ensemble of fine-tuned VGGNet architectures trained on 3 separate databases: ImageNet, Places MIT, and Food101, which is expected to capture the diversity of the dataset and combine the strengths of the different models at predicting labels which correspond to its training data (e.g., Places MIT for restaurant’s interior and exterior, Food101 for food, etc). The outputs of the 3 models can be fused by max pooling or averaging the probabilities- assigning label = 1 or label = 0 for each class’ neuron depending on probability > 0.5 or not.

Since we are also using Yelp data, we are interested in how they modify their learning architecture to specialize on the dataset. Their training data consisted of 2000 restaurants and ~230,000 images. Each image belonged to a restaurant, and each restaurant was labeled with one of 9 categories: {good for lunch, good for dinner, takes reservations, outdoor seating, expensive, alcohol, table service, ambience is classy, good for kids}. A particular challenge, which also exists in our project, is that instead of having labels for every image, there is just one set of labels for each restaurant. There is no information about individual images, except for matching it with associated restaurants.

The paper’s first approach was to assign the labels of each restaurant to all its images. Transforming the problem to a binary relevance problem, the authors found that using CaffeNet to extract features from the images and then Scikit-Learn to train a SVM for each label gave “quite good” results. However, they abandoned the restaurant → image labeling method because it assumes every image reflects every label, which is obviously not true.

The improved baseline used a method that assigned a feature vector to each restaurant. Image feature extraction was first done by extracting the first fully connected layer of CaffeNet pre-trained on ImageNet, which is the most general of all fully connected layers but still benefited from previous convolution layers. Restaurant features were constructed by averaging feature vectors for all its images.

Improvements to baseline were made by experimenting with different layers, architecture, pooling and classification techniques. The authors investigated whether it is better to use more specific or more general layers (fc6 vs fc7 vs etc). They also tried VGGNet as it is substantially deeper than CafeeNet but not as computationally expensive or complex as ResNet or GoogLeNet, respectively. Max pooling was investigated instead of averaging for restaurant vectors. Finally, a multi-layer perceptron was used instead of 9 different SVMs, to include relations between labels. This was more relevant to us because a SVM is for classification and our project is a regression problem, so we would use a fully connected layer if we wanted the end result to be a rating rather than probabilities of different classes.

After optimizing the above, the authors focused on different datasets to pre-train or fine-tune a CNN on. The idea was to extract more expressive features for certain types of images; e.g by using a model used for food when extracting features for a food image rather than relying on ImageNet which is trained on a very general image database. The final result was 3 fine-tuned VGGNet models trained on ImageNet, a places database (MIT’s Places) for scene recognition and a food database (Food-101), to be more representative of Yelp images. Six fusion methods were investigated, the best was found to be late fusion, specifically bagging, which uses a voting procedure such that at least 2 models have to output probability > 0.5 for a class for the ensemble to predict that class. The model achieves an F1 score of 0.82, which is 12% higher than baseline (0.73) and obtained 25th of 355 participants in the competition.

**More Prior Work[5][6]:**

**User Modeling with Neural Network for Review Rating Prediction**

One question we investigate in our project is how image-based approaches compare with (user-based) text approaches. This paper proposes a novel neural network method for rating prediction (1-5) that incorporates user-specific modifications to words through a user matrix that premultiplies word vectors, followed by a document composition vector model (DCVM) that formulates the representation of each review and finally a softmax layer for rating classification. The model beats several baseline architectures like VecAvg[5], RAE[5], PVDM[5] and CNN[5], and can therefore be treated as state-of-the-art for text-based rating prediction.

**A Collaborative Neural Model for Rating Prediction by Leveraging User Reviews and Product Images**

This paper presents a novel rating prediction model (NRPTV) by jointly modeling text features from user reviews and visual features obtained from product images. It is useful to compare performance of our image-based model with this to see whether images alone are a relatively good predictor of rating compared to textual + visual information, addressing the main question outlined in the “Introduction”.

**Technical Methods**

Given that we propose a novel input space—namely, images—for rating prediction, it is paramount that appropriate technical methods and formulations are used. We seek to first formulate an architecture that is capable of both image processing and regression to a linear rating scale from one to five stars. Additionally, we must have a means of deploying this architecture.

We propose preliminary, high-level architecture choices that are known to perform well for the objectives outlined above. To begin, convolutional neural networks are an effective architecture for image processing [7] that we will leverage. Though deep CNNs are most popularly used in computer vision tasks for object detection, classification, and segmentation in a wide variety of fields [8], the architecture has also shown great levels of efficacy in audio- [9] and text-based [10] tasks. Interestingly, we observe that CNNs are capable of sentiment analysis over both mediums to a high degree of performance [9, 10].

The foundations for additional architectural specifications for our CNN take inspiration from similar previous works that analyze data sourced from consumer review platforms—albeit, these papers take a more conventional approach through the use of metadata and text reviews to train their architectures. One work in particular by Tang et al. describes an effective neural network method that predicts Yelp and Rotten Tomatoes star ratings from text reviews. Text reviews are fed into a novel sentiment analysis pipeline that produces a vector representing the overall sentimental representation of the text. This sentiment vector is then used as an input into a final review prediction framework [11].

The applications of CNN architectures to sentiment analysis, its ubiquity in vision tasks, and the sentiment-to-rating pipeline above give us a reasonable degree of confidence that a CNN-based methodology will be successful. Tang et al.’s work also provides an attractive starting point for our image-based rating prediction architecture. In particular, the text-to-sentiment substructures detailed above can be replaced with some CNN-based image substructure that produces an analogous vector representing the image in sentiment or semantic space. Indeed, sentiment analysis of images can be performed through CNNs [12] as opposed to the more commonly observed semantic labeling of images [13]. Whether a sentiment-based, semantic-based, or hybrid approach is most optimal is unclear. Additionally, the comparative performance and adequacy of a generic CNN architecture for our desired image-to-rating prediction is also an important subject of exploration. Moving further from baselines, the work by Joris Baan [4] provides an attractive alternative, adopting a more advanced CNN-based ensemble method, with a similar idea of using the power of (fine-tuned) CNN architectures as feature extractors.

We will make use of the popular python-based machine learning package PyTorch to build our architectures. Notably, PyTorch has a broad set of vision tools in its torchvision library which includes models for semantic segmentation [14]. Further flexibility is also provided through custom neural network modules [15]. These two features will prove to be essential for realizing the formulation provided above. Additional information will be documented through plotting packages such as matplotlib.

**Anticipated Results**

In general, our team expects good performance from the network; i.e good generalization of restaurant rating from a set of images. As outlined in the Technical Methods section, CNNs are strong for image processing and sentiment analysis, so our baseline CNN linear regression model that uses image as input and output rating is expected to perform well.

We set a realistic lower bound as the results obtained from the authors of [2]. This is because the models used in their study were simple and not tailored towards the specific dataset; so, our CNN baseline should perform as well as their linear/logistic regression models due its higher expressive power. Furthermore, we can play with the complexity of our network to better match the dataset, e.g., by following the ensemble method of [4]. Finally, our input data is assumed to be more representative of restaurant quality- images of food and restaurant interior/exterior are expected to be more reflective of the success of a business than metadata like prices, age, etc [3].

Given we improve our network complexity, e.g., to the level of [4], we expect better results than baseline. The ensemble method gave good results for classification into a subset of 9 labels in a multi-label setting. Our problem is simpler as it is a regression problem with just one output- a numeric value from 1-5. Thus, we expect it to be easier for our model to learn relationships between hidden vectors and final ratings, assuming they exist.

Lastly, we expect to predict appropriate ratings more frequently for restaurants where ground truth occurs frequently than extreme ratings. This is because the training set will have much more restaurants with ratings around 3 than with ratings 1 or 5. So, our model may learn to predict ratings for these restaurants, and perform poorly for extreme cases due to a small number of training examples. We hope to counter this by dataset augmentation techniques like rotations and blur, but this will only capture variations in camera and not diversity in image content.

**References**

[1] <https://www.yelp.com/dataset/documentation/photos>

[2] <http://cs229.stanford.edu/proj2017/final-reports/5244334.pdf>

[3] <https://deliverypdf.ssrn.com/delivery.php?ID=819027008027064117091126024019066092037036069049083071005107005093090113119030097093096050033032121023027005081022116005124125023073034069016010102111099104027007084021049034097022112005080093111105107120067121094120119024108071111126003002006071020088&EXT=pdf&INDEX=TRUE>

[4] <https://staff.fnwi.uva.nl/b.bredeweg/pdf/BSc/20152016/Baan.pdf>

[5] <https://www.ijcai.org/Proceedings/15/Papers/193.pdf>

[6] <http://yongfeng.me/attach/airs17-chen1.pdf>

[7]<https://link.springer.com/content/pdf/10.1007/s13244-018-0639-9.pdf>

[8] https://link.springer.com/article/10.1007/s13748-019-00203-0

[9] https://www.sciencedirect.com/science/article/pii/S1746809421005437

[10] https://ieeexplore.ieee.org/document/8398173

[11]<https://www.ijcai.org/Proceedings/15/Papers/193.pdf>

[12] <https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9556/9270>

[13]<https://arxiv.org/pdf/2106.11596.pdf>

[14] https://pytorch.org/vision/stable/models.html#semantic-segmentation

[15] <https://pytorch.org/tutorials/beginner/examples_nn/two_layer_net_module.html>