**Restaurant Rating from Food Images: README file**

This drive contains the following files/directories.

**Data Scraping and Processing:**

This folder contains the following files, related to data collection and processing.

**Data Extraction and DataLoader Construction for YelpNet**

Pytorch's DataLoaders let us define large datasets that can be processed from files as inputs to our model. Data loading is GPU parallelizable, which is a necessity given the size of our dataset. This file is used to construct the data loaders we use in training.

**Get Rating Yelp Dataset Food**

The Yelp dataset contains 40,000 images labeled as “food” along with the associated business IDs. This script makes calls to the Yelp API based on the business ID to retrieve the rating of the restaurant and up to 3 text reviews of the restaurant. Also, contains the web scraping code to collect full length reviews for each restaurant

**Yelp and Google Testing**

A relatively unimportant file at this point. Used to test making calls to the Yelp API and the Google Maps API if needed in the future.

**Text Reviews Preprocess**

This file extracts a reviews dataframe that contains text reviews and associated positive (>= 3.5) or negative (<= 3) labels.It is used to train a network that generates custom embeddings.

**Documents**

This folder contains all final submission documents and rough notes

**Final Architectures**

This folder contains files and subdirectories related to our proposed VGH-Net architecture.

**BERT Embeddings**

This notebook implements a pretrained BERT model to generate embeddings from text

**Text to Embedding**

This file uses the dataframe from *Text Reviews Preprocess.ipynb* to train a network to predict sentiment from text. We plan to test its success compared to BERT embeddings by evaluating VGH-Net’s accuracy if we use this network instead of BERT embeddings

**Yelp Final Architecture 1**

First attempt at implementing VGH-Net. Contains bugs related to memory leaks

**Yelp Final Architecture 2**

Final implementation of VGH-Net using all 40,000 images in the dataset

**Yelp Final Architecture 3 (PRUNED)**

Implementation of VGH-Net using pruned dataset. The training procedure is the same, but we prune the dataset by removing reviews with ratings in the range [3,5], so that our dataset is more balanced.

**Debugging, Copies, Misc**

Initial attempts at implementing and training VGH-Net components (CNN and FCN). These can be ignored when looking at final submission, but are stored as they were useful to refer to when working on the doc that eventually served as our final notebook.

**LongReviewScrapingJSONs**

This folder contains in-progress attempts to web-scrape full length reviews for each restaurant in the Yelp Dataset

**foodEntriesWithLongReviewsA1.json**

A json file with 11,752 full-length reviews from the first 1,207 restaurants.

**Long Review Scraping JSON Documentation**

A document tracking the indices of the restaurants that have been successfully scraped.

**Model Exports**

This folder contains saved losses and weights for our final architecture on different training sessions.

**Losses:**

This folder contains saved losses for all training iterations

**Weights:**

This folder contains saved weights for all training iterations

**Export Version tracker:**

This file contains details for each of the training iterations. This includes specifying text batch size, image batch size, FCN learning rate, CNN learning rate and outlining differences between training iterations.

**Old Architectures**

This contains preliminary architectures we explored before exploring a novel architecture. It contains the following subfiles/subdirectories:

**Simple CNN:**

Implementation of simple CNN for Yelp Rating Prediction

**More Complex CNN Architectures**

Implementation of hyperparameter tuning on simple CNN

**PreTrain Testing:**

Notebook for learning how to use a pre-trained network, and testing on toy dataset

**Pretrained/Fine-tuned:**

This folder contains implementations of fine-tuned VGG16, ResNet50 and MobileNetv2 architectures, as well as the ensemble network

**Object Detection Tutorial**

This file can be ignored. It may be interesting in the future to explore an architecture that first detects objects from images, then performs regression. The motivation behind this idea is similar to the current motivation behind our proposed two-step process (see report); we want to observe the effectiveness of an architecture that takes the images and gives a different representation of them, then uses this representation to produce a rating.

**ECE324 Final Project Presentation**

This file is the slides for our presentation

**foodEntries.json**

.json file with food entry information. There is no need for all team members to download all 40k images because we can just mount the drive on google colab and use the json file to build a dataframe.

**foodEntriesWithReviews,json**

Similar to foodEntries.json, but all only contains data for food images with associated reviews.

**foodPhotos.zip**

This is a zip folder with the 40,000 images labeled as “food” from the Yelp Photo Dataset.