

Multi-Label Multi-Instance Classification of User Submitted Yelp Photos

Sreekanth Krishnaiah, Kashish Kothari, Siddharth Chakravarthy
University of Illinois, Urbana-Champaign

1 Executive Summary

As part of completing the 4-credit portion of CS 498, the team entered a Yelp Restaurant Photo Classification Challenge on Kaggle. The project challenged the participants to build a model that automatically tags restaurants with multiple labels using an underlying dataset comprised of user-submitted photos. The task of predicting attributes/labels for restaurants using user-submitted photos is a type of Multi-Instance Multi-Label learning (MIML) classification problem. The instances here are *images* and a set of instances represent the *images belonging to a restaurant to which we must attribute labels to*.

Convolutional Neural Networks(CNNs) are generally used for multi-class or multi-label classification. Our project encompasses the classification of a set of multiple instances or 'bags' with multiple labels instead of the classification of a single instance. This requires us to think beyond conventional CNNs and modify them for MIML classification. We worked on some interesting upgrades to the CNN model, to achieve the goals of the project. Some of the methods include:

- **Model 1:** Incremental Principal Component Analysis(IPCA) with Logistic Regression and Support Vector Machines(SVMs)
- **Model 2:** Convolution Neural Network for Image level classification
- **Model 3:** Inspired by [2], where the authors take mean and normalize the filters before computing the neuron responses, we adapted this approach to develop a loss function that uses a mean/max layer in CNNs to enable learning of multi instance loss
- **Model 4:** Inspired by [8], we extract features by training CNN with modified loss functions and then use SVM to predict attributes of restaurants
- **Model 5:** Use Vector Quantization for MIML classification

We use the *Mean F1 Score* as the evaluation metric to evaluate the performance of our model. Our analysis indicates that **model 2**, **model 3** and **model 5** which use all instances/images of a restaurant to calculate loss and train the classifier, give superior F1 scores. CNN with SVM performed the best with a F1 score of 0.75. An example of final classification results have been tabulated in Figure 1.

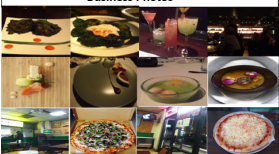


Business Photos	Correct Labels	Predicted Labels
	1,2,4,5,6,7	1,2,4,5,6,7
	1,2,4,6,7	0,1,2,4,6,7
	0, 8	3,8

Figure 1: Example of classification results

2 Introduction

Apart from writing restaurant reviews on Yelp, users can also upload photos highlighting their experiences. At the time of uploading photos and writing reviews, users have the option of attributing the restaurants with certain labels such as whether the restaurant is expensive or takes reservations or has outdoor seating. However, tagging restaurants with labels isn't necessarily mandatory for the users and hence Yelp relied on user uploaded photos to find a model that would automatically label restaurants that describe various features of a restaurant.

The problem we are trying to solve in this project is an example of MIML classification. In normal classification tasks, we would have a single instance, usually an image, which would then be classified to a class or multiple classes. This is called multi-class or multi label classification problem. While in multi-class, the classes are mutually exclusive, in multi-label classification, the training set is composed of instances each associated with a set of labels, and the task is to predict the labels of unseen instances through analyzing training instances with known label sets.

However, in our case we have multiple instances (images) or bags belonging to a restaurant and each restaurant has multiple labels associated with it. We need to design a classifier that trains on the bag of images where each image is a feature vector and predict the labels of new restaurants that contain a new set of images in their bags. Since we are dealing with images, we turn to Convolutional Neural Networks (CNNs) and modify them to deal MIML classification.

Overall, we are presented with a two-fold problem in this project. First, extracting restaurant level features and second, designing a modified CNN to enable MIML classification. In the first method, we extract image feature vectors through IPCA and perform logistic regression and SVM to obtain baseline F1 scores. In the next few methods, we explore various CNN models for MIML settings. The second method involves training a Deep CNN (DCNN) from scratch and then extracting the weights from one of the fully connected layers. These weights, which are feature vectors for images are then aggregated across restaurants to obtain restaurant level features and are trained using an SVM. In the third method, we train a DCNN with a mean and max layer before applying cross-entropy loss. Here we aggregate the feature vectors of images that belong to a restaurant after the fully connected layer and then calculate the multi- instance loss. We also use a pretrained VGG network to compare the F1 scores with our CNN models.

2.1 Related Work

The first step towards using deep CNNs for image classification came up in 2012 when the AlexNet achieved a top-5 error rate of 15.6% in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC)[3]. The CNN architectures evolved over time and became more robust in their prediction, which this was evident through the top-5 error rate of 6.7% of the GoogleNet in 2014 and the ResNET that achieved a top-5 error rate of 3.6% [4,5]. Zhou et al [6] proposed MIML classification using MIMLBOOST and MIMLSVM. The former predicts each label independently while the latter assumes instances belonging to a bag contribute equally. While both MIMLBOOST and MIMLSVM performed better than many existing algorithms at the time it was proposed, we can take advantage of Convolutional Neural Networks to help achieve better performance.

Very recently, Lingyun Song et al. [7] proposed a deep-modal CNN for MIML (MMCNN-MIML) classification. By combining CNNs with MIML learning, their proposed model represents each

image as a bag of instances for image classification and inherits the merits of both CNNs and MIML. Its advantages include automatically generating instance representations for MIML by exploiting the architecture of CNNs and uses the label correlations by grouping labels in its later layers. This model incorporates the textual context of label groups to generate multi-modal instances, which are effective in discriminating visually similar objects belonging to different groups.

2.2 Dataset, Features and Pre-Processing

The data set has been provided by Yelp on Kaggle. It consists of a map of business IDs to labels, a map of business IDs to photos, and the photos themselves. The photos have varying resolutions. The training dataset consists of 2000 businesses and about 230000 photos amounting to close to 13GB of data. Due to computational constraints, we trained the models only on 200 business IDs which amount to close to 1GB and use 100 business ids for validation and test sets. This amounts to about 26000 photos for training and 10000 and 12000 photos for validation and testing respectively. There are 9 possible labels that each business can be categorized into, described by the table in figure 2.

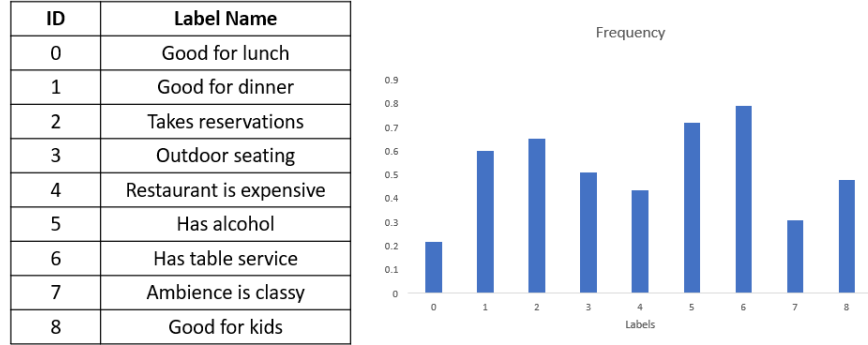


Figure 2: Description of Labels and Frequency of Occurrence In The Training Set

We have encoded the labels in the form of one hot encoding vectors of dimension 1x9 where each dimension is either 1 or 0 describing the business feature attribute. Furthermore, we pre-process the images in a manner where the all the images are randomly resized and center cropped to make them of size 227x227. The implementation was computationally intensive and was executed on Vectordash where the instances used an NVIDIA GTX 1080Ti with 16 GB RAM.

The metric used for the model validation is the F1 score which is defined by the formula given below.

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

3 Implementation Details

A total of **4 models** were implemented. All models had the same training, validation and test datasets.

As a general introduction to the problem; during the model-training phase, the input to the CNN

is a set of labelled data points $\{X_i, Y_i\}$, where $i = 1 \dots N$, where each $X_i = \{x_{i1}, x_{i1} \dots x_{ik}\}$ is essentially a set of k_i instances, where each of the specified instances could be different for each of the N training samples. Ultimately, this would lead to a correspondence to an entity where each label Y_i represents the ground truth set. $Y_i = \{y_1, y_2 \dots y_n\}$, a set with n attributes.

Figure 3 illustrates the Deep CNN architecture implemented in our project. This is a baseline architecture and all the models discussed below for the MIML classification utilize a modified version of this architecture.

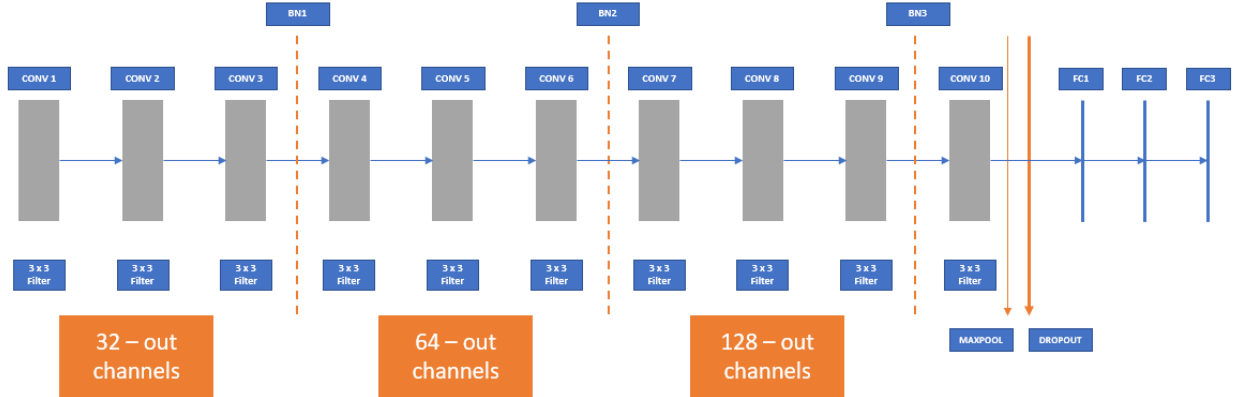


Figure 3: Model Architecture

As the figure illustrates, the model architecture incorporates 10 convolution layers, 3 fully connected layers, 1 max-pool layer, 1 dropout layer in addition to having 3 batch norm layers.

3.1 Model 1: IPCA with Logistic Regression and SVMs

The first method involves extracting image features using IPCA and then applying Logistic Regression and SVMs. The F1 score obtained from these methods will serve as our baseline F1 score. Although Kaggle suggests random guessing for baseline scores, we thought logistic regression and SVMs would be a good place to start.

For this baseline method, we will attribute restaurant labels to the images directly. In the first step of the pipeline, we extract one-dimensional feature vectors of all the images of size 227x227 in the training set. Each image is of size 227x227x3 and converting them into one dimensional feature vectors results in 150528 features. We applied IPCA, a dimensionality reduction technique which reduces the number of highly correlated features, in our case, refers to pixels, without loss of variance. In this way, we decrease the number of correlated pixels for each image and the resultant smaller image pixel can contribute more to the final decision describing about 90% variation from the existing image pixels. After IPCA, 600 principal components were finally chosen.

In the second step of the pipeline, we group several instances of image feature vectors belonging to the same business together and average the features. For example, let's say for each business,

we obtain an $N \times 600$ feature matrix where N is the number of images belonging to the restaurant, we take mean along the first dimension, to obtain a 1×600 feature vector for that business. This results in 200×600 and 100×600 feature matrices for the train and test splits respectively. For both logistic regression and SVM classification, we use a one v/s all classifier.

SVM (using a linear kernel) performed slightly better than the logistic regression. SVM tries to transform dataset into a rich feature space and separates it using maximum margin and support vectors while logistic regression tries to optimize log likelihood function with sigmoid probabilities. Overall evaluation of the model on the test data set gave us a satisfactory F1 score which is used as the baseline score but there was a lot of scope in improving the model and result.

3.2 Model 2: Convolution Neural Networks

Model 1 was not capable of learning complex features from the images. The solution to this is utilizing a CNN model. The CNN takes the convolutions of images and provides relevant image features and hence performs extremely well in comparison to a vanilla logistic regression and SVM. Thus we construct a simple baseline CNN as described in the figure 4 , as our next model. In this method, the CNN uses a dataset with photos directly tagged with labels. The CNN has 9 neurons in the final fully connected layer representing each of the classes. The CNN model was trained on every image in the training data set with each of the images having their own individual class labels. Multi-label softmax loss was calculated based on every image label while training the parameters.

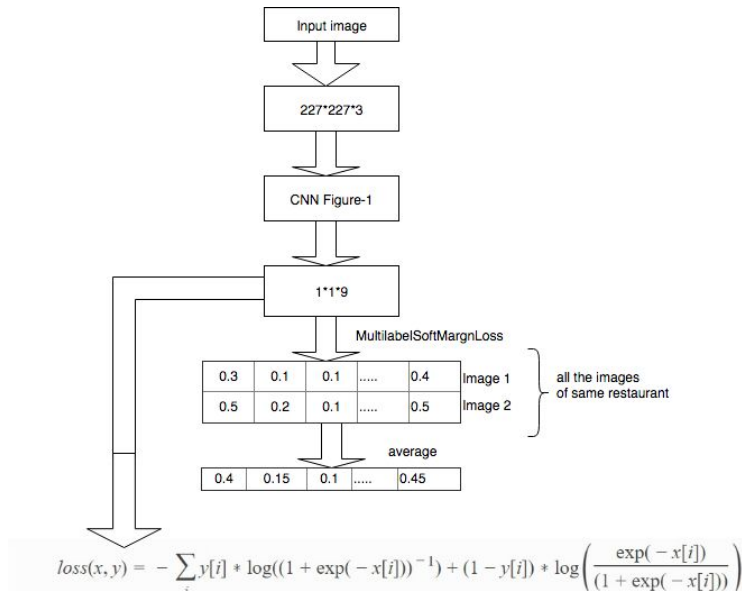


Figure 4: Baseline CNN Model

For training the CNN, the following hyper parameters were used. The whole model was trained over 120 epochs at a learning rate of 0.001 using an Adam optimizer. For pre-processing, data augmentation techniques like the random transfer and random crop were performed.

As this is MIML classification, we are not classifying images but instead we are classifying the business. So we merged the output layer features after the loss function from all the images

belonging to the same restaurant and took average of it along its first dimension. A brief overview of this method is illustrated in figure 4. The average class probability features are then converted to binary form with prob > 0.5 as 1 else 0. Lastly F1 score is calculated for each business based on their encoded labels in the test set.

3.3 Model 3: CNN with Customized Mean/Max Loss Layer

The Convolutional Neural Network we built in model 2 is a model that is most suitable for a single instance vanilla classification problem where we treat each photo independently during the training phase. The loss is calculated for a single instance and the weights are backpropagated along each image vector. We only aggregate the output from the loss layer for each business after the CNN training is completed. However, in the models built this way, we will lose information of how certain pictures are grouped into a same business and share the same labels. This information can be valuable for prediction, because photos of a same business may have similar background features, and the meaning of 9 labels are all concerned with the whole environment and vibe of restaurants. Hence, we modified the CNN model to include a mean/max layer in between the fully connected layers and a loss function that aggregates multiple instances belonging to a business ID and calculates a multiple instance loss which is then backpropagated. The architecture is illustrated in figure 5.

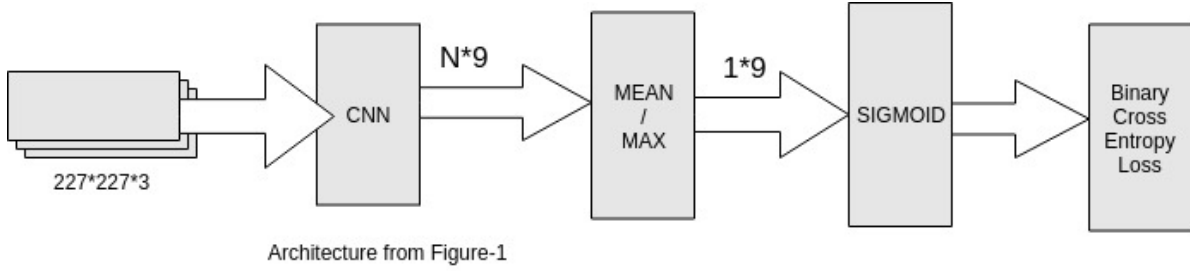


Figure 5: Architecture of the CNN with a customized mean/max layer

In the model which employs a mean layer, say the output from the FC3 layer is of size $N \times 9$ where N is the number of images of the restaurant and 9 is the number of attributes. Lets call this activation X^i for the i^{th} instance. The mean layer basically takes the mean of all the instances belonging to the restaurant along the first dimension and calculates X^{i*} which is of the dimension $1 \times N$. This final activation is passed through a sigmoid layer after which cross entropy loss L_i is calculated which is given by the equations described below. Y_i is the actual class label.

$$L_i = Y^i \log\left(\frac{e^{X^{i*}}}{1 + e^{X^{i*}}}\right) + (1 - Y^i) \log\left(\frac{e^{X^{i*}}}{1 + e^{X^{i*}}}\right)$$

$$X^{i*} = \sum_{k=1}^N X_k^i$$

In the model which employs a max layer, we take the max of all instances belonging to a restaurant along the first dimension after the fully connected layer. We then pass it through a sigmoid function and proceed to calculate the cross entropy loss similar to the procedure used in the case of the mean layer.

3.4 Model 4: Convolutional Neural Network with SVM

The next model we implemented is adapted from the work done by Xiao-Xiao Niu et al. [8] which utilizes the synergy of two superior classifiers: CNNs and SVMs. In the model we implemented, CNN works as a trainable feature extractor and SVM performs as a classifier.

The first phase of this method involves extracting features from a Deep CNN trained from scratch. We use the same architecture as model 3, which has a mean layer between the final fully connected layer and the loss function. This aggregates the features vectors of all instances belonging to each business along its first dimension and enables multi instance training. Once the model is trained, we extract the train and test image feature vectors from the second fully connected layer. Each image feature vector has 600 dimensions after the fully connected layer.

In the second step of the pipeline, we group several instances of image feature vectors belonging to the same business together and average the features. Say for each business, we obtain a $N \times 600$ feature matrix where N is the number of images belonging to the restaurant, we take mean along the first dimension, to obtain a 1×600 feature vector for that business. This results in a 200×600 and 100×600 feature matrices for train and test respectively. These features are then trained using a linear kernel SVM one vs rest classifier.

3.5 Model 5: CNN with Vector Quantization

Vector quantization is a general compression technique in the field of signal processing. We have adapted this approach in order to cluster the image features based on their class labels. Image features are extracted from the CNN model trained from scratch. In the second phase, we apply vector quantization on the extracted image features. At the time of training, we use kmeans for clustering all the train image features belonging to the same class labels. There are 9 class labels and each label can take binary values 0,1 and therefore a combination of $2^9=512$ binary values. This results in a total number of $2^9=512$ clusters.

Every cluster group is represented by its centroid feature. In the testing phase, we calculate euclidean distance between the test image feature and all the 512 cluster centroids. The test image is attributed to that cluster and label which yields the shortest distance. Lastly, we calculate F1 score and measure the performance of the model.

3.6 Transfer Learning(VGG) with mean layer

Finally, we wanted to compare our model performance to the pretrained models. Hence we built a model using pretrained VGG network. This model is similar to the CNN with mean layer model, except here we use a pretrained VGG network instead of our own CNN.

4 Results and Analysis

Figure 7 shows the tabulated results of various models we explored. Along with the F1 scores for each of the labels, overall F1 scores for each of the models have also been displayed. While it isnt surprising to see the models involving pre-trained VGG performing better than CNN models trained from scratch, it is worth noting that CNN with SVM and Vector Quantization models perform equally good followed by CNN model with mean layer.

Model	F1 Score of Various Labels									Overall F1 Score
	0	1	2	3	4	5	6	7	8	
IPCA + Logistic	0.4	0.51	0.56	0.47	0.4	0.62	0.65	0.38	0.63	0.53
IPCA + SVM	0.37	0.53	0.63	0.45	0.30	0.69	0.79	0.18	0.67	0.57
Basic CNN	0.55	0.67	0.72	0.65	0.58	0.71	0.79	0.57	0.78	0.67
CNN + mean layer	0.66	0.75	0.82	0.61	0.60	0.72	0.79	0.66	0.81	0.71
CNN + max layer	0.53	0.62	0.65	0.65	0.55	0.72	0.79	0.58	0.80	0.65
CNN + SVM	0.65	0.81	0.81	0.62	0.65	0.76	0.81	0.51	0.9	0.76
VGG + mean layer	0.58	0.85	0.88	0.67	0.74	0.80	0.92	0.68	0.87	0.78
VGG + max layer	0.45	0.85	0.89	0.63	0.74	0.80	0.85	0.75	0.81	0.75
Vector Quantization	0.47	0.89	0.87	0.57	0.59	0.86	0.89	0.71	0.75	0.75

Figure 6: F1 scores of all discussed models

We first perform IPCA and extract 26000 train and 10000 test image feature vectors of 600 dimensions each. Using logistic regression and SVM to perform one-vs-rest classification on train and test matrices is computationally expensive. Even after training SVM for about 48 hours, we still had no result in sight. Hence, we average multiple instances belonging to a business along its first dimension. This also helps the model learn better as images belonging to a business share the same information and helps generalizing the image features. The F1 scores obtained from Logistic Regression and SVMs are 0.53 and 0.57 respectively which are simple yet competitive baseline scores. The baseline score in the actual competition on kaggle is around 0.41. We observe that labels 0,4 and 7 perform miserably as compared to other labels.

We built a CNN model from scratch by attributing the labels of the businesses to its constituent images. In this case, the loss is calculated on each image and the model doesn't take into consideration all the multiple instances belonging to each business while training. Only after training are the output feature vectors of each business aggregated. The scores are given in the third row of the table. This model gives a superior score of 0.67 and performs well on all the labels as compared to the IPCA-SVM model. While we see improvement over baseline performance, this model doesn't consider the multiple instances of a business in the training phase. We only calculate a single instance loss and back propagate the weights. Hence there is a scope for improvement if we could modify the CNN model to include multiple instances of a business in the training phase and then calculate loss over multiple instances of a business.

We accomplish this in the next model where we employ a mean layer between the final FC3 layer and the sigmoid layers achieving an F1 score of 0.71. The cutoff probability chosen for the outputs of the last layer to be classified as 0 or 1 is 0.37. We tried different cutoff probability values to classify the output as seen in figure 7 but highest F1 score was obtained at 0.37. This model considers multiple instances of each business in training phase itself and hence outperforms the simple CNN single instance classification model. One thing to note is that this model gives superior F1 scores for all labels except label 3 when compared to the single instance CNN model. In contrast, a similar model where a max layer was used instead of mean layer results in an F1 score of 0.65 which is worse than a single instance CNN model. It also gives lower F1 scores across all the

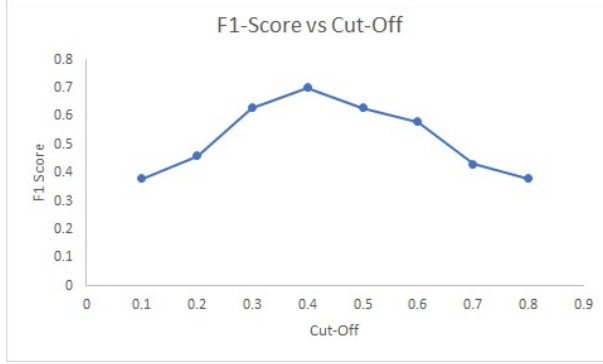


Figure 7: Cutoff probability vs f1 score

labels. This might be because rather than using all the images of a business, this model only uses one feature vector i.e. the maximum feature vector across the first dimension. In the next method, we use the same CNN with mean layer model described above but this time extract the features of size $N \times 600$ from the FC2 layer for both train and test images and then perform classification using one-vs-rest SVM, where N is the number of instances in each business. This uses the best of both CNN and SVM and considers multiple instances of each business. This method gives the best accuracy of 0.76 and performs well across all the labels.

To compare how our CNN models performed, we use a pre-trained VGG model for comparison. In this case, we use a VGG along with a mean layer in between the fully connected and sigmoid layer just like the CNN-mean layer model described above. This gives an F1 score of 0.78.

We train our fifth model combining the concepts of vector quantization and CNN. The results obtained from this method is an F1 score of 0.75 which is very close to our best model CNN + SVM. In this process, we consider all the combinations of labels possible i.e 512 unique cases. This becomes a problem of classifying images into 512 different classes. Such granular clustering based approach has helped to achieve the F1 score very similar to our best model.

5 Discussions

Based on the individual class levels, it can be inferred from figure 7 that all the models predict the "good for kids" class with the highest confidence. On the other hand, the "outdoor seating" class has been predicted with the lowest confidence given that it has the lowest F1 score amongst all classes. A reason for the low confidence could stem from the fact that the outdoor seating (as illustrated in figure 8) didn't have distinct feature patches making it difficult for the models to learn.

Among the experimented models, the model comprising the VGG+mean layer was the best one for predicting the "outdoor seating" class. In transfer learning, the model is pre-trained on a lot of image objects. Therefore it was able to identify those distinct object features for "outdoor seating" class that the CNN + SVM model failed to learn from the training images. This proves that transfer learning outperforms the CNN model trained from scratch for predicting certain classes. "Good for kids" and "has table service" classes were among the well predicted classes. One of the reason could be that "Table Service" class was present in 68 % of restaurants in the training data. An interesting observation that the team made was with regards to the "Good for kids" class. This



Figure 8: Images labeled as Outdoor Seating Class

is because there are no sufficient distinct features to identify if a restaurant is good for kids, with such high precision and recall but still all the models performed well for this label.

6 Conclusion

As part of this project, the team implemented multiple models and assessed the robustness of prediction of the said models by comparing their F1 scores with that of the transfer learning approach which incorporated the VGG framework. In summary, the model that incorporates the CNN and SVM has the most superior performance with an F1 score of 0.76.

The robustness of this prediction can be attested to the fact that there only exists a 2% difference in the F1 score of this model and the model that uses transfer learning within the VGG framework. For this project, the team worked on randomly subsetting data of the original dataset. With more computational resources, the team can explore the resultant F1 scores of having gone through all of the data.

7 Statement of Individual contribution

Everyone contributed equally to the project. The following are the specific parts each of us worked on:

Sreekanth Krishnaiah: CNN with mean and max layer, CNN with SVM, VGG with mean and max layer, Data Preprocessing, Project report

Kashish Kothari: CNN, CNN+vector Quantization, Data Preprocessing, CNN Preprocessing (Sampler, Data Loader for all the models in Pytorch), Project report

Siddharth Chakravarthy: Data preprocessing and exploration, IPCA+SVM, IPCA+Logistic, Project report

References

- [1] <https://engineeringblog.yelp.com/2015/12/yelp-restaurant-photo-classification-kaggle.html>
- [2] G. Papandreou, I. Kokkinos, and P.-A. Savalle, Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3903-3911.
- [3] Krizhevsky, Sutskever, Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", *Proceedings of the 25th International Conference on Neural Information Processing Systems*. - Volume 1, NIPS'12, 2012, pp. 1097-1105.
- [4] Szegedy, Liu, Jia, Sermanet, Reed, Anguelov, Erhan, Vanhoucke, Rabinovich, "Going Deeper With Convolutions", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.
- [5] He, Zhang, Ren, Sun, "Deep Residual Learning for Image Recognition", *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [6] Z.-H. Zhou and M.-L. Zhang. Multi-instance multi-label learning with application to scene classification. In *NIPS*, 2006.
- [7] L. Song et al., "A Deep Multi-Modal CNN for Multi-Instance Multi-Label Image Classification," in *IEEE Transactions on Image Processing*, vol. 27, no. 12, pp. 6025-6038, Dec. 2018. doi: 10.1109/TIP.2018.2864920
- [8] Xiao-Xiao Niu and Ching Y. Suen. 2012. A novel hybrid CNN-SVM classifier for recognizing handwritten digits. *Pattern Recogn.* 45, 4 (April 2012), 1318-1325. DOI=<http://dx.doi.org/10.1016/j.patcog.2011.09.021>
- [9] <https://www.kaggle.com/enerrio/data-exploration-yelp-classification>