Multiple Instance Multi-Label Learning for Yelp Restaurant Photo Classification

Jade Huang
Stanford University
jade.huang@stanford.edu

Abstract

Users upload nearly as many photos reviews as they write restaurant reviews on Yelp. At review time, they have the option of commenting upon attributes of the restaurant such as whether the restaurant is good for lunch or whether it has a classy ambience. We use various approaches with convolutional neural networks to label businesses with attributes using the photos of the businesses in this task of multiple instance multi-label learning. The data, provided by Yelp and Kaggle as part of the Yelp Restaurant Photo Classification Challenge, consists of over 200,000 training images and 1996 training businesses as well as over 200,000 test images and 10000 test businesses, with a varied distribution of labels over businesses.

While attempts to simplify the task down to single instance multi-label learning did not prove fruitful, utilizing transfer learning with CNNs pretrained on ImageNet followed by an SVM Classifier resulted in a 0.79 F1 score, a 58% increase from a random guesser baseline. Labels that proved hardest to predict included "good for lunch" while the best performing label was "has table service". Future work could include finetuning the transfer learning and SVM classifier pipeline, exploring model ensembles, and experimenting with other pretrained networks.

1. Introduction

Besides writing restaurant reviews on Yelp, a site where users can review businesses, Yelpers upload photos indicative of the restaurants they are writing about. For example, patio seating indicates the presence of outdoor seating. A bar area would hint at the restaurant serving alcohol. A picture of a sizable portion of food in daylight could indicate the restaurant is good for lunch. Typically, users have the option of manually entering in these fields at review time, but it is not compulsory. As a result, some restaurants are left partially categorized or even un-categorized.

Rather than rely on a user who may not manually fill in all the labels at review time, the goal is to utilize user-uploaded photos to automatically label restaurants with descriptive attributes for the benefit and convenience of not only users but also restaurants.

The input to our algorithm is two-fold: businesses represented by IDs and their associated images. We first use a convolutional neural network to extract features from the images. Then, for each business, we represent the associated business as a conglomeration of its images, for example, by taking the mean of all related image features. We then train a linear SVM classifier to output a predicted vector of labels for each business.

This problem is an example of multiple instance learning meets multiple label classification. Instead of having labels for every data point as in vanilla supervised classification, we have labels for sets of instances or "bags". Instead of classifying a single data point with a single class as in vanilla multi-class classification, we classify a set of instances with vector of binary labels. Here, our instances are images, where sets of images represent a business which has a label vector.

2. Related Work

2.1. Multiple Instance Multi-label Learning

Zhou et al. [14] proposed two ways of taking the multiple instance multiple label (MIML) problem: MIMLBoost and MIMLSVM. MIMLBoost assumes the labels are independent, thus predicting each independently. MIMLBoost also assumes that all instances in a bag contribute independently and equally to the label of the bag, MIMLSVM considers the relationships between each bag through clustering at the bag level and decomposes the multi-label learning problem into multiple binary classification problems. It is clever to cluster the different bags as knowing that two bags are similar could aid in predicting the labels, however depending on the size of the dataset, this could be computationally expensive. Though the MIML algorithms proposed outperformed other state-of-the-art algorithms, especially for the scene classification dataset of images, a convolutional neural network perhaps could take advantage of the the image data and perform even better.

In contrast, Cheplygina et al. [2] represent each bag by

a vector of its dissimilarities to other bags and treats the vector of dissimilarities as a feature representation. While such an approach performs well, the dissimilarity function is unique to the distribution and size of a given dataset.

2.2. Multiple Instance Learning with Convolutional Neural Networks

Kraus et al. [4] experimented with utilizing global pooling layers with fully connected layers in a convolutional neural network to not only learn relationships between instances of the same class, but also to learn relationships between classes. This was done using an adaptive Noisy-AND pooling function which activates a bag level probability once the mean of instance probabilities reach a certain threshold. Strengths of this approach are how the network is trainable end-to-end, as Kraus et al. utilized a non-linear back-propagation approach in multiplying the pooling activation for each class, as well as a learnable threshold of the Noisy-AND pooling function for each class. A weakness that the authors note is that the dataset likely includes mislabeled samples, and the total number of samples, though augmented with various different crops, was not large: one dataset used contained about 20 images per class.

Another convolutional multiple instance learning approach was proposed by Pathak et. al [7] where the task is to learn pixel-level semantic segmentation through instance-level labels signaling the absence or presence of an object in the task of weakly supervised image segmentation. In the case of this paper, the bag-level label is the image label, whereas each instance is a pixel. While the modified and finetuned VGG 16-layer net used results in a 96% relative improvement over the baseline, the predictions are larger and more vague than the ground truth. While for our dataset we only have bag-level labels, by passing our instances through award-winning convolutional neural networks pretrained on ImageNet, we can essentially recover instance-level labels with relatively high confidence of accuracy.

2.3. Multi-label Learning with CNNs

Wei et al. [13] proposes a model to predict multiple labels for images using a CNN pretrained on a large-scale single-label dataset such as ImageNet, aggregating predictions from different object segment hypotheses using max pooling. It is interesting that max pooling is used to produce the multi-label predictions and not average pooling, as max pooling could possibly discount various hypotheses whereas average pooling would take into account all hypotheses.

2.4. Transfer Learning

The results of Razavian et al. [9] suggest that using features extracted from deep learning with convolutional nets to use as image representations results in competitive results compared to highly tuned state-of-the-art systems in visual recognition tasks using a wide variety of datasets. Shin et. al [11] also explore cross-dataset transfer learning using AlexNet [5] and GoogLeNet [12] with medical image datasets, finding that as the complexity of the CNN model increases, so does the accuracy level. Both Razavian et al. and Sermanet et al. [10] propose a CNN-SVM pipeline when using transfer learning. While the off-the-shelf CNN representations perform well used simply as feature extractors, finetuning could possibly increase performance even more. However it is promising that with no finetuning, feature extraction provides such a boost in a wide variety of image tasks.

3. Methods

3.1. Single Instance Multi-label Learning

The first proposed algorithm is to decouple labels from businesses and assign the labels to each business' associated images, and feed the images through a pretrained network with a modified output layer, finetuning based on the gold labels for images rather than the gold labels for businesses.

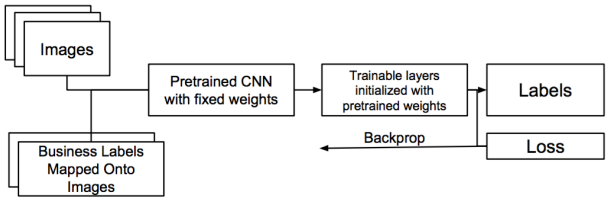

Figure 1. Mapping business labels to images and using them to finetune a pretrained network with modified end layers.

This is not a perfect assumption, as one cannot assume that all photos of a business are indicative of all labels, for example, a restaurant good for lunch and dinner may have two photos validating one label each, but the lunch photo is not indicative of dinner and the dinner photo is not indicative of lunch.

The reasoning is that since a neural network is made up of matrix operations, the aim is to create a trainable end-to-end network without worrying about having to map businesses to images during the process of training.

Since this is a case of multi-label learning, we utilize a cross-entropy objective function by which we base our loss and gradient updates on.

$$-\sum_{labels} \vec{t} * log(\vec{p}) + (1 - \vec{t}) * log(1 - \vec{p}) \qquad (1)$$

where p represents the predicted vector of labels and t represents the target vector of labels.

In the network itself, we scrap out the typical softmax output layer for a sigmoid layer followed by a custom threshold layer as the output layer. Since the output of the sigmoid layer is in the range of [0,1], we interpret these numbers as probabilities of each label. The threshold layer rounds the probabilities from floats to 0 or 1, resulting in an output of a binary vector corresponding to each of the nine labels for each image.

Once training has completed, we take the mean of predicted labels over all related images to represent the labels for a particular business. The values are then rounded to their nearest integer. This method was implemented using a combination of Nolearn [6] and Lasagne [3].

3.2. MIML with Transfer Learning + SVM

The second proposed algorithm is to utilize transfer learning paired with a linear Support Vector Machine Classifier in an instance of multi-instance learning. The "Food" category in ImageNet has 1495 subcategories and 1,001,000 images in total. The subcategories include beverage, dish, course, wheat, milk, and more. The dataset provided by Yelp and Kaggle, in comparison, comprises of 234,842 images, which is 23.46% the amount of food images in ImageNet. The food images in ImageNet extremely similar to the images Yelp users upload with their reviews, which influenced the decision to utilize transfer learning.

We fed images through networks pretrained on ImageNet until the penultimate layer (or even the layer before that) so that instead of a softmax probability over all ImageNet classes, we obtained a "code" vector for each image. In the case of VGG CNN-S, we obtained a 4096-dimensional code for each image when extracting from the sixth and seventh fully-connected layers, and for GoogLeNet, 1000-dimensional or 1024-dimensional code for each image when extracting from the penultimate or third-to-last layer of the network.

Having obtained these codes or features for each image, we must then relate the images to their associated network by merging the relevant image features together. We experimented with taking the mean and max of relevant images to represent the relevant business. By taking the mean, we thus represent each business by its "average" image feature. By taking the max, we represent each business by the maximum features over all of its images.

Having converted our image features into business features, we then can train an SVM classifier to predict 9-dimensional vectors for each business, where each of our nine labels receives a 1 or 0 score relating respectively to the presence or absence of a label. This is a case of multi-label learning, in comparison to multi-class learning where we predict one class out of multiple class options. We use a one-vs.-rest or one-vs.-all multi-label strategy of training one classifier per label, where the classifier in question is

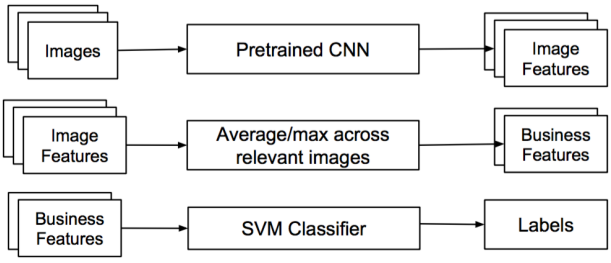

Figure 2. Three-stage transfer learning pipeline.

a linear SVM classifier. By using one classifier per label, the task is reduced to multiple cases of binary classifiers predicting an output of 1 or 0 for a given label for a given business.

For each label, we seek to find a hyperplane dividing positive examples (businesses with a certain label) from negative examples (businesses without a certain label). We seek to train the following linear mapping which gives us our score of whether a label should be present or not:

$$f(x_i, W, b) = Wx_i + b \tag{2}$$

where x_i is a given business feature vector, W is a matrix of weights, and b is a bias vector. The SVM utilizes the hinge loss in order to measure performance:

$$l = max(0, 1 - y \cdot \hat{y}) \tag{3}$$

When the predicted label $\hat{(y)}$ is equal to the actual label y, the loss is 0. When the predicted label is incorrect, the loss is 1. The SVM updates its weights and biases based on the gradient of the loss.

All methods were implemented with Lasagne [3] as well as the scikit-learn [8] library and were executed on NVIDIA GRID K520 GPU. The networks, VGG CNN-S [1] and GoogLeNet [12], and their pretrained weights were obtained from Lasagne's Model Zoo.

4. Dataset

Yelp provided a set of training images, test images, mappings from images to businesses for both the training images and test images via Kaggle, and labels for training businesses. The training dataset comprises of 234,842 training images and 1996 businesses, while the test dataset comprises of 237,152 images and 10,000 businesses.

In reality on Yelp in the "More business info" section, there can be up to 22 labels. For this competition with Kaggle, the set of labels was limited to a size of 9. The labels are described below:

While training the SVM Classifier, we utilized 80% of the training businesses to train on and the remaining 20% as a validation set.

Label #	Label Name
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

Table 1. A subset of Yelp's Business Info section used for the Yelp Kaggle competition as labels for businesses.

The images are user-uploaded photos in a variety of resolutions and sizes: some photos are in portrait mode, others in landscape. Some are shaped like a square, Instagram-style. To account for this variance in sizing, we performed data preprocessing in the form of resizing all images to 224 x 224.

When comparing images with the labels of their businesses, there are some labels a human can easily intuit from the image, and others not so much. When shown a picture of an alcoholic drink, it is easy for a human to say that the associated business has alcohol. A sandwich paired with good lighting could indicate a business is good for lunch. But some images are not so intuitive.

Figure 3. Training photos whose businesses are labeled as good for outdoor seating.

Figure 4. Training photos whose businesses are labeled as having table service.

Not all images are of food. There are images of cooks with children associated with an image that is good for dinner, has alcohol, and a classy ambience. There are pictures of menus, which are helpful when one wants to know what the business serves, but not so much if we are performing image classification and not paying attention to words in images. There are also images of certificates or awards, which similarly to images of menus, are helpful when one is looking for a good bite to eat.

Some of the labels are difficult to guess when shown a subset of photos from a business. Picking out that a business has outdoor seating is difficult when the majority of the photos are of food and not of the surrounding scenery. A similar problem arises when deciding if a restaurant offers table service. Most restaurants have tables, most images of food have a table underneath, but you do not know for certain whether a restaurant has table service unless you see a waiter. Many businesses with a classy ambience have photos that are poorly lit with neon lights, but this is true for all classy businesses.

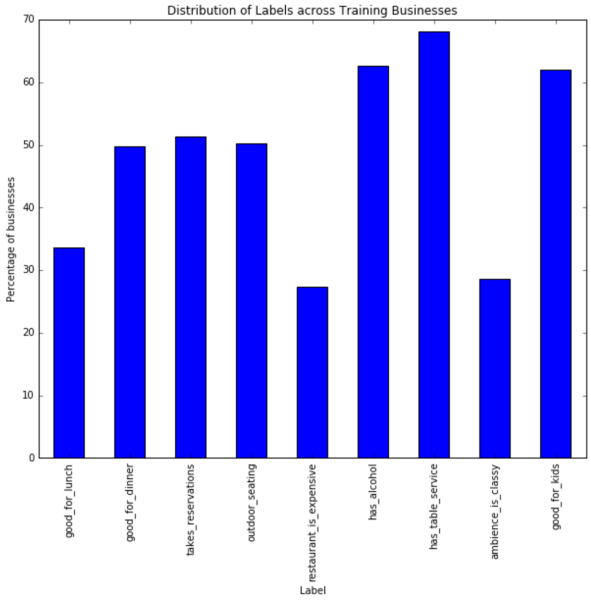

Figure 5. Distribution of labels across training businesses.

As we can see from the above chart, most of the training businesses, 68% in fact, were labeled as having table service. In contrast, only 34% of the businesses were labeled as being good for lunch. The two labels with less training data per say are restaurant is expensive and ambience is classy, but the performance for those two labels are higher by about 10% than the performance on good for lunch.

5. Experiments

5.1. Evaluation

The evaluation metric for this competition is the mean F1 score, which measures accuracy based on precision, the ratio of true positives (tp) to all predicted positives (tp + fp) and recall, the ratio of true positives to all actual positives (tp + fn).

$$F1 = 2\frac{p * r}{p + r} \tag{4}$$

where

$$p = \frac{tp}{tp + fp} \tag{5}$$

$$r = \frac{tp}{tp + fn} \tag{6}$$

5.2. Naive Baseline

Yelp implemented two baselines. The first is a naive guesser which makes a random assignment for each attribute with equal probability, resulting in a score of 0.4347. The second is sort of a nearest neighbor algorithm–it compares color distribution of all images of a test business and compares it to the average color distribution of businesses with positive attribute values and negative attribute values respectively, assigning the value with a more similar color distribution to the test business. The resulting score is 0.6459.

We implemented a version of Yelp's random guesser baseline in Python which generates nine random numbers from a uniform distribution and then thresholds: numbers greater than 0.5 are thresholded to 1 otherwise to 0. This random guesser, similarly to Yelp's version, achieves 0.41337 F1 score on a validation set of 400 businesses and 0.41520 score on the test set of 10000 businesses.

5.3. Single Instance Multi-label Learning

5.3.1 Smaller Neural Networks

Before utilizing a pretrained network, we wanted to examine the performance of smaller neural networks as an additional baseline measure on a smaller portion of the data. For this task, the training set is composed of around 4000 images representing 100 businesses while my validation set is composed of 1000 images representing 50 businesses. Business labels were mapped to images.

We implemented two smaller networks: a small neural network of fully connected layers and a convolutional network of convolution layers, dropout layers, pooling layers, and fully connected layers. A small network composed of two fully connected layers can achieve higher than guessing randomly with 0.49 accuracy and a convolutional network with the below architecture can achieve around 0.63501 accuracy with minimal hyperparameter tuning and 5 epochs. For both of these small networks, a small subset of the data was used: 2000 images with a 60-40 training-validation split.

These results are in line with expectations–both networks perform better than guessing at random. A neural network without convolutions performs better than guessing at random, but a convolutional network taking advantage of the image structure performs better than a neural network without convolutions.

Layer #	Layer Name	Size
0	input	3x32x32
1	dropout	3x32x32
2	conv8-3	32x30x30
3	conv8-3	32x28x28
4	pool2-2	32x14x14
5	dropout	32x14x14
6	hidden	50
7	dropout	50
8	hidden	9
9	output	9

Table 2. Architecture of small convolutional neural network used.

Architecture	F1 Score
Random Guesser	0.41337
2-layer FC NN	0.49
Small CNN	**0.63501**
VGG CNN-S + Finetuning	0.60881

Table 3. Baseline results on respective validation sets

5.3.2 Transfer Learning

We then utilized a VGG CNN-S network pretrained on ImageNet with a modified output layer for multi-label learning with all layers except for the fully connected layers fixed so that the last several layers could be finetuned for the dataset.

Hyperparameter search for learning rate and regularization was done randomly and cross-validated utilizing an 80-20 split over the training data. Adam was utilized for the update function and the mini-batch size, due to GPU memory constraints, was 50.

After training for one day over 80% of the training data, while the training loss decreased steadily, the accuracy could not achieve higher than a 0.55667239 training score and 0.60880853 validation score.

This result, though disappointing, is expected as by mapping business labels to images, we have many very different images with the exact same labels which may not even accurately represent all the labels, thus making it hard for the classifier to generalize and perform well. In addition, the higher score of the smaller CNN by 3% can be explained in part by that a smaller dataset was used to test the smaller CNN in comparison to the VGG CNN-S network.

5.4. MIML with Transfer Learning + SVM

We utilized two CNNs pretrained on ImageNet, courtesy of Lasagne: VGG CNN-S and GoogLeNet. For both networks, we experimented with extracting features from the penultimate layer as well as the layer before the penultimate layer. We refer to these layers as "FC7" and "FC6"

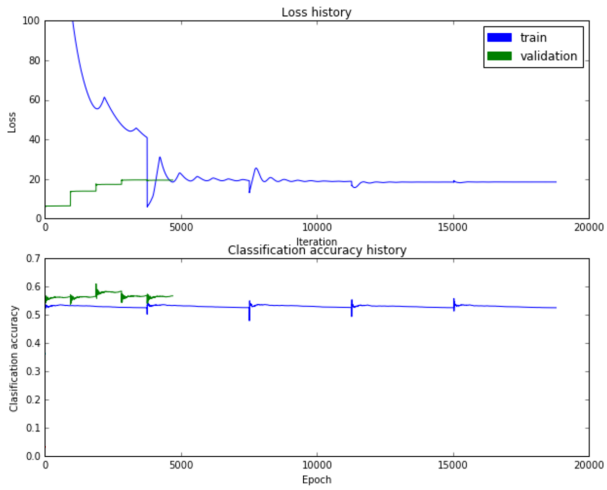

Figure 6. A decreasing training loss coupled with a mysteriously increasing validation loss and stagnant accuracies when experimenting with converting the problem into a single instance multi-label learning problem.

respectively throughout this paper.

Extracting from VGG CNN-S from both the FC7 and FC6 layers resulted in a 4096-dimensional feature vector for each image, whereas extracting from GoogLeNet from FC7 resulted in a 1000-dimensional feature vector and from FC6 a 1024-dimensional feature vector. For all networks, we utilized a mini-batch size of 100 as to not run out of GPU memory.

We additionally experimented with taking the mean of the relevant image feature vectors to represent a business versus taking the max of the relevant image feature vectors to represent a business. Taking the mean of image feature vectors to represent a business is intuitive as it can be seen as representing a business with its average image. Doing so incorporates all features from all relevant images in an equally weighted fashion. By contrast, taking the max of image feature vectors to represent a business is not as intuitive, but can be seen as representing a business with the representative features. Another motivation is related to max-pooling layers in convolutional networks: typically max-pooling is used over mean-pooling due to better performance.

In comparing the F1 score when taking the mean versus taking the max, the scores are comparable, however for GoogLeNet the F1 score decreases for both layers. This drop in performance is as expected as instead of taking into account all images, by taking the max, we ignore some images, and thus perhaps some clues which would point to us saying yes to a certain label.

What is promising that the F1 score on the test set, which as revealed by Kaggle is only done on 30% of the complete test set, is not too much lower than our validation F1 scores,

Model	Layer	Mean	Max	Test
VGG CNN-S	FC7	0.7908	**0.7972**	0.75442
VGG CNN-S	FC6	**0.7934**	0.7905	0.75312
GoogLeNet	FC7	0.7882	0.7548	**0.76459**
GoogLeNet	FC6	0.7891	0.7593	0.74650

Table 4. F1 Scores on validation set and 30% of test set for each model-layer combination comparing taking the mean of image features to represent business features to taking the max of image features to represent business features. The mean image feature was used for the test set submission.

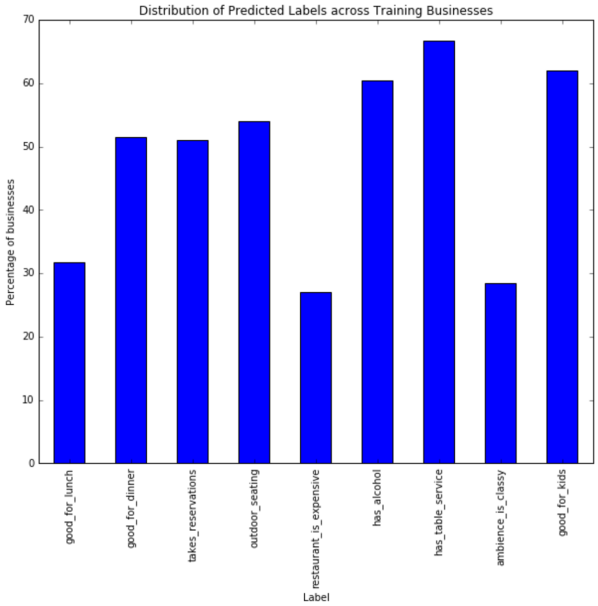

Figure 7. Distribution of predicted labels using image features extracted from VGG FC 7 and taking the mean of relevant image features to represent a business feature.

suggesting that our models were not prone to overfitting.

When utilizing the mean of relevant image features, the predicted distribution of labels amongst all classifiers were proportionally close to the the true distribution. VGG CNN-S at FC7 underguesses on "takes reservations" while VGG CNN-S at FC6 better captures the proportions. A similar phenomenon occurs with GoogLeNet where features at FC6 result in a more accurate distribution of labels than at FC7.

On an individual label basis, the four SVM classifiers perform the best at predicting "has table service" and perform the worst when predicting "good for lunch", "restaurant is expensive", and "ambience is classy", which was contrary to the thinking that "outdoor seating" would be the hardest to predict. Though, "outdoor seating" typically scored the next worst after the three aforementioned labels, so our intuition wasn't completely off.

Figure 8. Photos from a restaurant tagged as being good for lunch that our SVM Classifier trained on features extracted from GoogLeNet FC6 thought were not so good for lunch.

Why would it be the hardest to predict what is good for lunch? One would think that a sandwich, salad in daylight would definitely point to a business being good for lunch. But some restaurants are good for both lunch and dinner, and perhaps have more dinner photos, which tend to be more dimly lit. Such an example is shown above.

Recall that the classifier was trained on the mean image feature, so a business was represented as the features of the average image, per say. The features were extracted from networks that were pretrained on ImageNet, which has many classes relating to food and identifying food. It is surprising that "has_table_service" performed the best, at almost 90% accuracy. Perhaps it is a mix of most restaurants having table service, food dishes always being on a table-like surface, and having the most data regarding restaurants having table service.

6. Conclusion

For the task of multiple instance multi-label learning with Yelp business photos, we found that the highest performing algorithm was that of transfer learning with an SVM classifier. Extracting image features from a pretrained network such as VGG CNN-S or GoogLeNet, taking the average over a business' set of images, and training an SVM classifiers to predict upon the business improved upon the baseline by 58%. Converting the problem into a single instance multi-label learning problem was not as successful, potential reasons being that not every photo from a given business will represent all the labels of the business, thus a classifier has trouble generalizing and learning.

For future work with more time, more team members, and more computational resources, we would like to explore finetuning the transfer learning + SVM pipeline, model ensembles, and other pretrained networks. An additional fun comparison could be to explore the performance of extracting features from the mean image of a business to represent a business as opposed to our approach of using the mean image feature to represent a business.

References

[1] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *CoRR*, abs/1405.3531, 2014.

[2] V. Cheplygina, D. M. J. Tax, and M. Loog. Multiple Instance Learning with Bag Dissimilarities. *ArXiv e-prints*, Sept. 2013.

[3] S. Dieleman, J. Schlter, C. Raffel, E. Olson, S. K. Snderby, D. Nouri, D. Maturana, M. Thoma, E. Battenberg, J. Kelly, J. D. Fauw, M. Heilman, diogo149, B. McFee, H. Weideman, takacsg84, peterderivaz, Jon, instagibbs, D. K. Rasul, CongLiu, Britefury, and J. Degrave. Lasagne: First release., Aug. 2015.

[4] O. Z. Kraus, L. J. Ba, and B. J. Frey. Classifying and segmenting microscopy images using convolutional multiple instance learning. *CoRR*, abs/1511.05286, 2015.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[6] D. Nouri. Nolearn: Abstractions around neural net libraries, most notably lasagne.

[7] D. Pathak, E. Shelhamer, J. Long, and T. Darrell. Fully convolutional multi-class multiple instance learning. *CoRR*, abs/1412.7144, 2014.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[9] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. *CoRR*, abs/1403.6382, 2014.

[10] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013.

[11] H. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. J. Mollura, and R. M. Summers. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *CoRR*, abs/1602.03409, 2016.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[13] Y. Wei, W. Xia, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan. CNN: single-label to multi-label. *CoRR*, abs/1406.5726, 2014.

[14] Z. Zhou, M. Zhang, S. Huang, and Y. Li. MIML: A framework for learning with ambiguous objects. *CoRR*, abs/0808.3231, 2008.