Final Project Report

Yelp Restaurant Photo Classification by Machine Learning

Kai Ye

ye000064@umn.edu

Abstract

This project aims to predict multiple attribute labels of restaurants with photos submitted by users from Yelp. HOG and pre-trained Inception CNN are used as feature descriptors. Image features are integrated using SMI algorithm and EMD algorithm for each business respectively. Multiple-label classifiers are built using a combination of SVM, LR, and RF models with the average probability adopted. The threshold for classification is set as 0.47. Based on the simulation of this process, the project compared different feature descriptors in image recognition and several strategies in solving MIMLL problems. The Inception CNN with SMI algorithm performed best in the experiment and achieved an F_1-score of 0.8036, which indicates an effective label matching scheme for MIMIL classification problems like the Yelp restaurant photo classification problem.

I. Introduction

The idea of this project comes from a competition on Kaggle.com [1]. Since eating is one of the essential requirements of human beings, finding a good restaurant is always one of the best options to relax after work or study. Yelp app is one of the most widely-known apps to search for foods and restaurants. The restaurants on Yelp usually are labeled with nine general attributes: "Good for lunch", "Good for dinner", "Take Reservations", "Outdoor seating", "Expensive", "Has alcohol", "Has table service", "Classy ambiance" and "Good for kids", which are all chosen by the Yelp community. Moreover, users are free to upload photos and write reviews on the restaurants they visited. The popularity of smartphones makes it easier to takes pictures of food and share it on the Yelp community. These pictures consist of the menu, the ambiance, the food, the shopfronts, selfies or group photos with friends, and so on. It's useful to label the restaurants with appropriate attributes based on these user-submitted photos, which helps other users find their target restaurants faster and more efficiently. On the other hand, it also helps filter out unrelated or duplicate photos, which highlights effective reviews and gives more suitable views about restaurants.

II. Problem Description

Photos from several restaurants are given by Yelp in both train and test sets. The train set consists of around 234842 photos from 2000 restaurants. The test set consists of 234872 photos from 10000 restaurants. The task is to build a machine learning model to classify these photo-related restaurants with different attributes. To match appropriate labels from pictures taken by users, the problem can be solved in two parts.

A. Feature Extraction

Pictures are too abstract to be analyzed directly by computers. Effective image processing methods are required to transform pictures into distinct structured data, which are called features extracted from the pictures. These features represent the intrinsic properties of the original images and these properties can be used in the classification process then. There are several ways to extract features from images, such as non-deep feature descriptors like Histogram of Oriented Gradients method (HOG) [2], Scale Invariant Feature Transform (SIFT) [3], and deep feature descriptors like Convolutional Neural Network (CNN) [4]. Depending on the method adopted, the features extracted from the same image are different. The performance of the classifiers also varies based on the features used in training.

B. Feature Processing and Classification

After the features are extracted from the pictures, automatic labeling using machine learning can be conducted on these extracted data. The pictures are uploaded by customers from several restaurants. In other words, each restaurant in the data set is related to different numbers of pictures. Also, each restaurant is labeled with one or more tags. Thus this problem is a multiple-instance multiple-label learning problem (MIMLL) [5]. The differences between MIMLL and conventional machine learning problems are shown in Figure 1.

Conventional supervised learning MIMLL

Figure 1. The difference in learning frameworks between traditional learning and MIMLL (Adapted from Figure 1 in Ref [6])

Multiple-instance (MI) indicates that the features extracted from multiple photos should be merged into a single feature for the corresponding restaurant. Using the given mapping list from photo ID to business ID, the number of photos corresponding to restaurants is plotted in Figure 2 for the train set and the test set (only around 20% photos in the list are given in the test set).

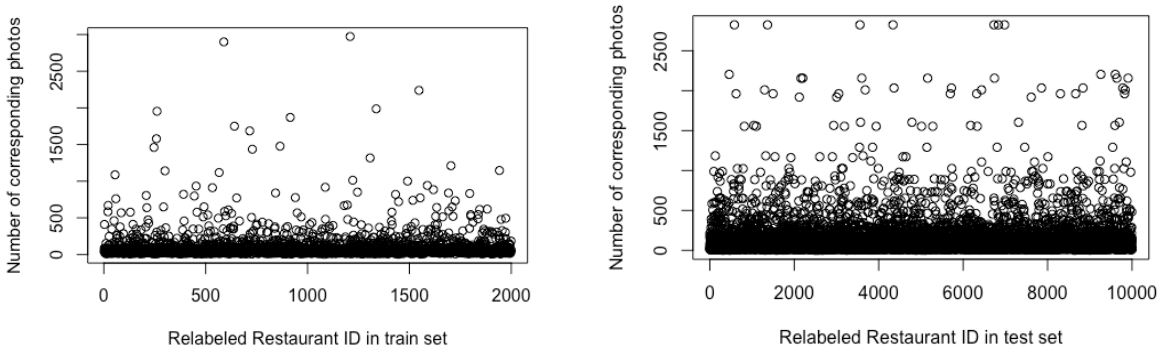

Figure 2. Number of corresponding photos for each restaurant

The corresponding number of photos ranges from 1 to around 2500 in both sets, indicating this ML dataset is unbalanced. Thus it's fundamentally important to utilize appropriate algorithms to integrate multiple features of photos into a single feature for the restaurant. It might also be useful to exclude restaurants with very few photos since the integrated feature might be misleading.

Multiple-label (ML) means the classifier should be trained based on several labels, which is different from the conventional binary classifiers. The distribution of different labels in the train set is analyzed and plotted in Figure 3. The nine labels are tagged using number 0 to 8 for convenience in the following simulations. The numbers of restaurants in each label bin are in similar size and no obvious correlations between these labels can be seen from the distribution.

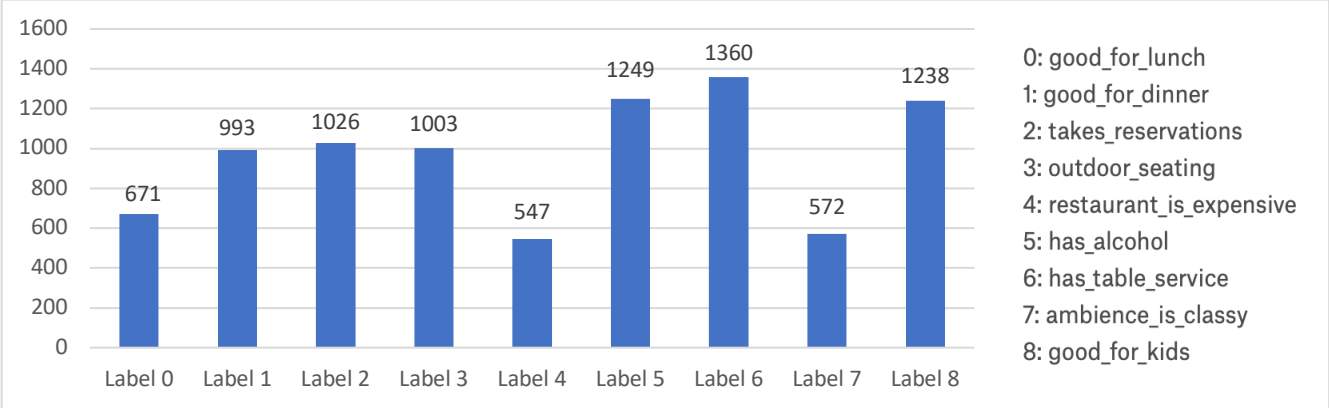

Figure 3. Number of restaurants for each class

III. Prior Work

Initially, it is thought that the problem is based on food image recognition. As for food image recognition, H. Hoashi et al. [7] proposed a food image recognition system for 85 food categories by feature fusion with a 62.52% classification rate. However, this problem is different from conventional food recognition since the labels are attributes of restaurants rather than food categories and the images are not only about food. In other words, this problem is simpler with fewer classification categories while more complicated in image recognition since these user-submitted photos are informative. These photos may give us information about the restaurants in various dimensions such as the environment inside or outside, the food categories, the customer categories and so on. Thus an effective feature extractor plays an important role in enhancing the performance of automatic labeling. They adopted feature fusion of various kinds of image features in their food recognition system. 17 features consisting of bag-of-features(BOF), color histogram, Gabor texture features, and HOG are extracted and integrated with uniform weights for training.

However, these features are all somewhat "shallow" that represent direct intrinsic attributes of the pictures in the scale of color, the edge of shapes, etc. These shallow feature descriptors use statistic methods such as histograms to transform the image features that are straightforward in visualization and human-understandable into structured data.

Based on the research of Y. Kawano and K. Yanai [8] on food recognition, deep learning methods such as pre-trained CNN perform much better as feature extractors in image recognition problems. Since 2014 very deep convolutional networks started to become mainstream in computer vision and substantial gains are achieved on different benchmarks in various areas, such as the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [9]. Based on the GoogLeNet [10], which is the winning entry in ILSVRC 2014, C. Szegedy et al. brought up the idea of further improvement with the Inception architecture of GoogLeNet [11]. By adopting factorization into small and asymmetric convolutions, using auxiliary classifiers, and label smoothing as regularization, the image recognition accuracy of the Inception network is greatly improved with higher computational efficiency than its predecessors. Thus it has become popular and has been widely used in photo classification projects these years.

As for MIMLL problems, the review by J. Amores [12] indicates there are several methods in solving MIML and the performance varies on different data sets. He divides multiple-instance classifiers into three paradigms as shown in Figure 4: Instance space paradigm (IS), bag space paradigm (BS), and embedded space paradigm (ES). The main difference between these three paradigms is the information space in which the MI data is exploited. IS exploits the discriminative

information in each instance and aggregates local information, while BS and ES deal with bag-level information. The instances of different classes, the pictures from each restaurant in this problem, are treated as a bag of features instead of local individual instances. And each bag is treated as a whole entity with a focus on more global characteristics of the whole bag. BS implicitly extracts the bag-level information while ES performs explicitly by mapping the relevant information of the whole bag onto a single feature vector. And ES can be further divided into several categories based on different mapping strategies. The performance of IS, BS and ES paradigms varies on different data sets but the Earth Movers Distance (EMD) method in BS tends to be the best in all databases tested [12].

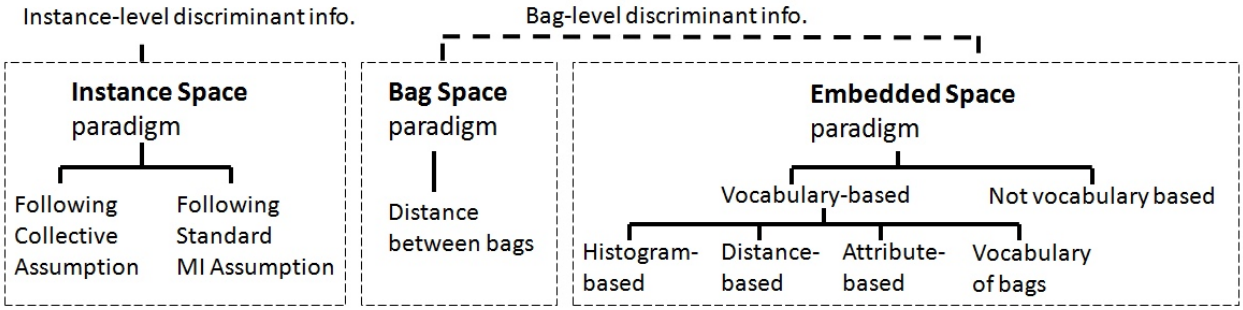

Figure 4. Proposed taxonomy of Multiple-instance classification methods (Adapted from Figure 4 in Ref [12])

ML is relatively easy to be analyzed. It can be solved by building multiple binary classifiers according to the number of labels. In each binary classifier, the label is fitted versus all of the others. This strategy is straightforward and interpretable since each label is fitted in one and only one conventional binary classifier. ML is resolved into multiple single-label classification problems. J. Read et al. [13] introduced another method to solve ML classification, by establishing classifier chains(CC). The "divide and conquer" strategy assumes that there are no interdependencies between label thus it can be partitioned into multiple independent problems. CC can model label correlations while maintaining the computational efficiency of the conventional binary relevance method. Instead of dividing and conquering, this chaining method builds N (N is the number of labels) classifiers in a linked chain through the feature space. In this method, the label information passes from the first classifier to the last one, thus the classification results are influenced by prior classifiers as shown in Figure 5. By using output y from the previous classifiers, this preserves the inter-label dependency of the labels. And conventional "dividing" method only uses the direct input x in training. However, the order of the classifiers in the chain can make a big difference. Thus there are some extended versions of CC that take all the possible orders into accounts such as Ensemble of CC (ECC) and Monte Carlo CC (MCC) [14].

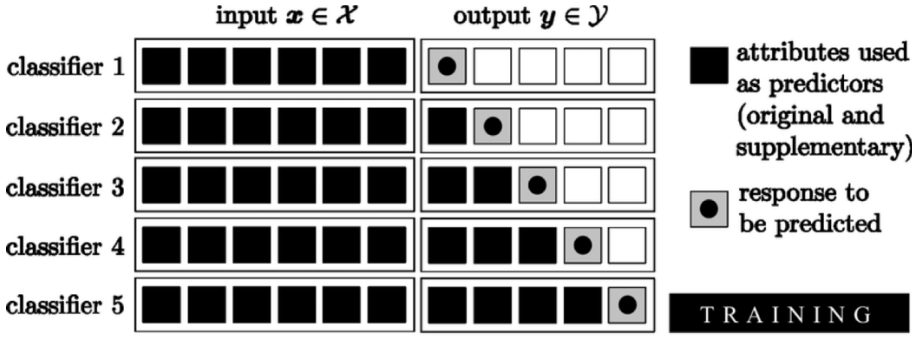

Figure 5. The framework of CC with 5 classifiers (Adapted from Figure 1 in Ref [15])

IV. Proposed Approach

A. Image features

In this project, both shallow and deep learning features are implemented for comparison.

a) Shallow feature extractor: HOG method

In this project, HOG is adopted to extract shallow features, as it's widely used for object detection in computer vision projects. Each photo is grayed and resized into 256×256 pixels and then partitioned into cells in the size of 16×16 pixels. The HOG descriptor is restricted in grayscale color space considering the workload, which may result in a slight decrease in the performance of the predictor according to the simulation by N.Dalal and B. Triggs [2]. The magnitude and orientation of the gradient are calculated for each pixel in the cell. Then the gradient data are summarized into a balanced frequency table dividing by 8 orientation bins, that is, a 1×8 orientation histogram of gradients is generated for each cell. Since there are 256 cells in each photo, the feature extracted for each photo would be a 1×2048 vector, which can be used in the training parts below. And this process can be visualized as Figure 6.

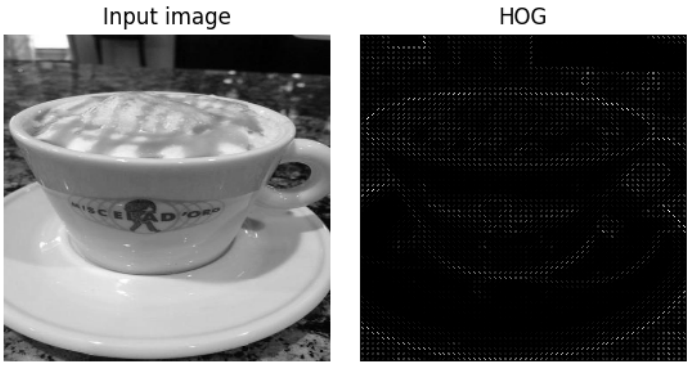

Figure 6. Original image compared with visualized HOG feature for Photo-944 in the train set

As shown above, the HOG feature is straightforward and interpretable. The shape of the coffee cup is shown by several small line segments representing the gradients. However, the gradients of the pixels are sensitive to the overall illumination. Since HOG is calculated in several small cells, some portion might be much brighter. To reduce the variance of illumination and shadowing, the process can be improved by normalizing the gradient histograms of cells over larger spatial regions, that is, so-called "blocks". Two choices of the block, rectangular blocks (R-HOG) and circular log-polar blocks (C-HOG), were introduced and compared [2]. And the effects vary based on different strategies of normalization. Generally, all methods showed significant improvement over the non-normalized data as shown in Figure 7. The miss rate nearly halves from 1×1 block to 2×2 block. And 3×3 blocks of 6×6 pixel cells perform best, with a 10.4% miss rate. While 2×2 blocks of 16×16 pixel cells are used in this project considering the large amount and the size of pictures.

The L2-norm scheme is used in this project and each block is set to 2×2 cells. Let v be the non-normalized vector containing all histograms in the block, then the L2-norm is given by $f = \frac{v}{\sqrt{\|v\|_2^2+\varepsilon^2}}$, where ε is a small constant to prevent singularity.

The HOG descriptor is implemented by "scikit-image" [16] in python and is utilized for the HOG processing in this project.

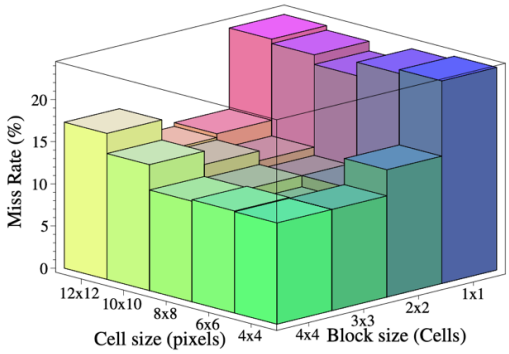

Figure 7. The miss rate at 10^{-4} False Positives Per Window as the cell and block sizes change. (Adapted from Figure5 in Ref [2])

b) Deep feature extractor: pre-trained Inception network by Mxnet [17]

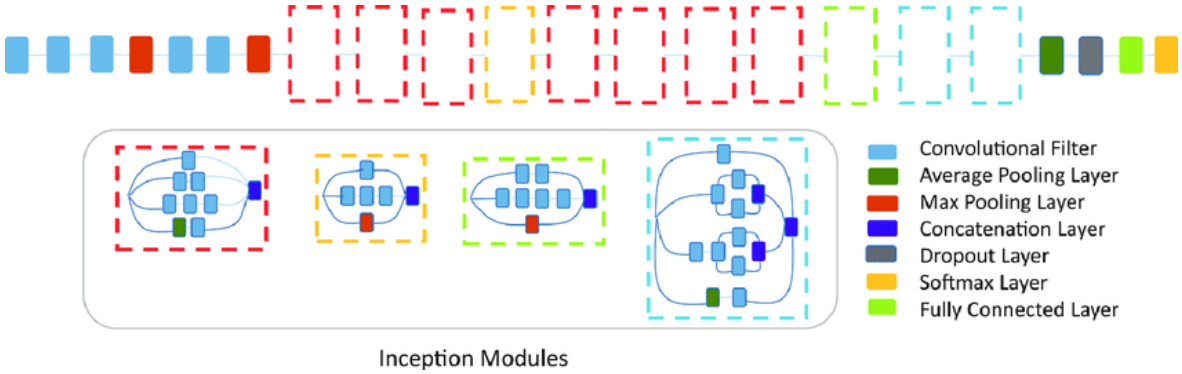

Figure 8. The architecture framework of Inception-v3 network (Adapted from Figure 3 in Ref [18])

Inception-v3 [11] loaded in Mxnet is adopted in this project and the schematic diagram of the Inception architecture is shown in Figure 8. The network is 48-layer deep and has 7 million parameters. It stacks 11 inception modules where each module consists of pooling layers and convolutional filters with rectified linear units as the activation function. The complexity of the structure makes it difficult to change the parameters and fine-tune the network. Thus it's directly used in this project. The input of the model is two-dimensional images resized to 299×299 pixels with RGB color channel. The output is a 2048 dimensional vector for each photo, which is the same size as extracted by HOG in this project.

B. Feature processing

As introduced in previous parts, there are several methods to solve the MI problem and two of them are selected to be implemented in this project for comparison.

a) IS paradigm: Simple MI algorithm (SMI)

The naïve attempt to solve MI problems is to transform them into traditional machine learning problems with a single instance. To achieve this goal, the simplest way is to take the mean of the feature of all the instances that correspond to the same business. In this way, the local individual information of instances is aggregated into the final single feature.

b) BS paradigm: The Earth Mover's Distance (EMD) algorithm

EMD is widely used in image retravel problem [19].In the BS paradigm, each set of features for a certain restaurant is treated as a "bag". To learn at the level bags of features, we can define a

distance function D(X, Y) that compares any two bags X and Y. Let $X = \{\overrightarrow{x_1}, \cdots, \overrightarrow{x_n}\}$, and $Y = \{\overrightarrow{y_1}, \cdots, \overrightarrow{y_n}\}$, then EMD is defined as: $D(X,Y) = \frac{\sum_i \sum_j w_{ij} \|\vec{x}_i - \vec{y}_j\|}{\sum_i \sum_j w_{ij}}$, where w_{ij} is weight coefficient obtained by minimizing $d(\vec{x}_i, \vec{y}_j) = \|\vec{x}_i - \vec{y}_j\|$ subject to the restrictions as the minimum cost flow problem [20]. The EMDs from one bag to all the other bags are calculated thus the integrated feature is a k-dimensional vector, where k is the number of restaurants in this problem. Since EMD includes an inner optimization process, there exist different choices to solve the minimum cost flow problem. The choice of method doesn't influence the final result and the network simplex algorithm [21] is adopted as it's the common way to solve this.

C. Classification using machine learning

Using the business feature achieved in previous steps and the given labels, the training of machine learning models is conventional. The final prediction is decided by taking the average probability of three classifier models: support vector machine(SVM), random forest(RF), and logistic regression(LR). The results that these classifiers generated indicate the probability of being in a certain class. By default of a binary classifier, since the result is either yes or no, the probability boundary is 0.5 by definition. While in practice, we can modify the threshold to a little bit higher or lower than 0.5 to get better results since usually the cases are not separable. Multi-label learning has already been implemented in the "scikit-learn" [22] package by using multiple independent binary classifiers. And it's reasonable to ignore the correlations among the labels in this problem. There are nine labels in this problem, so the multi-label classifier input is a nine-element vector of zeros and ones for each restaurant.

V. Results and Discussion

Firstly, some preprocessing on the data sets was conducted to avoid anomaly. Four duplicate and missing values were checked and removed. Then different combinations of algorithms are tested and compared.

It takes 4 hours to extract features from 0.24 million photos using HOG on a personal laptop with CPU only. And it takes 8 hours to extract features from 0.24 million photos using the Inception-v3 model loaded in Mxnet on a cloud GPU. Then the classifier is trained using 60% training data and the performance of the prediction on the rest 40% data is calculated using the F_1-score metric, which is given by $F_1 = \left(\frac{2}{recall^{-1} + precision^{-1}}\right) = 2 \cdot \frac{precision \cdot recall}{precision + recall}$. In this equation, the precision P is the number of correctly classified samples divided by the number of all classified samples returned by the classifier. The recall R is the number of correctly classifier samples divided by the number of all samples that should be in this class. F_1-score ranges from 0 in the worst case to 1 when both precision and recall are perfect.

Scheme \ Label	0	1	2	3	4	5	6	7	8	Overall
HOG + SMI	0.3953	0.7334	0.7839	0.5652	0.5820	0.8025	0.8613	0.5467	0.8306	0.7298
HOG + EMD	0.4631	0.7074	0.7573	0.5326	0.6352	0.7124	0.7937	0.5873	0.8011	0.7121
Inception-v3 + SMI	0.6863	0.7582	0.8547	0.6941	0.7582	0.8153	0.9261	0.7547	0.8619	0.8232
Inception-v3 + EMD	0.6578	0.7302	0.8419	0.6851	0.7124	0.8523	0.8542	0.7810	0.8312	0.8098

Table 1. Individual-class and overall F_1-score on 40% train set data

The results of the classifier on the train set are listed in Table 1 in the F_1-score metric. The threshold of the classifiers is set to 0.47 after several tests. And the data are plotted in Figure 9 for a better view. As shown in the figure, for the individual class F_1 scores, since nine binary classifiers are supposed to be independent, the prediction performance varies for different feature descriptors and MI algorithms. And there isn't a scheme that dominates in all the classes while Inception-v3 with SMI gives the best overall result. For the comparison between feature descriptors, it's obvious that the very deep features perform much better than the human-comprehensible shallow features. As for MI algorithms, the EMD algorithm improves the performance of some single-class classifier while the overall performance is decreased, this may result from the unbalanced number of instances (photos) between each bag (restaurant).

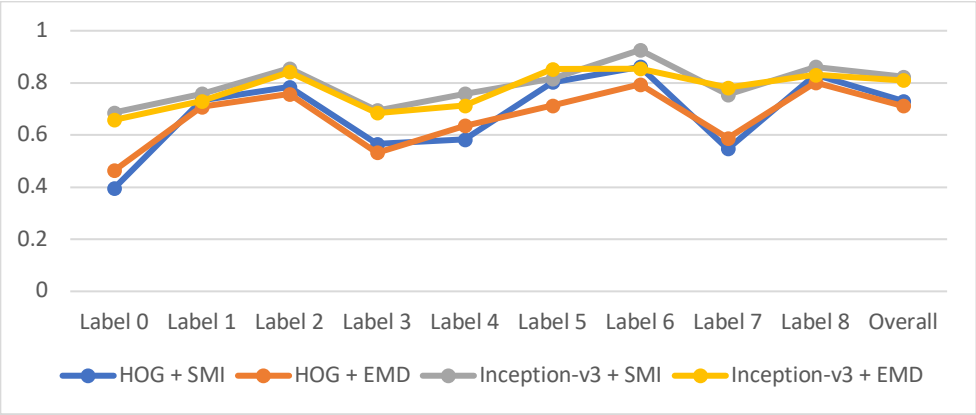

Figure 9. Individual-class and overall F_1-score on 40% train set data in the line chart

Then the classifier is trained using all the data in the train set to generate a prediction on the test set. The prediction results were uploaded to Kaggle and the performance measured in the F_1-metric is listed in Table 2. And the performance matches the conclusions we got from cross-validation on the train set.

Schemes	F_1-score on 30% of the test data	F_1-score on 70% of the test data
HOG + SMI	0.7107	0.7009
HOG + EMD	0.6988	0.6934
Inception-v3 + SMI	0.8113	0.8036
Inception-v3 + EMD	0.8013	0.7962

Table 2. Prediction performance of final models on the test set

Finally, the results of the classifier on the test set can be visualized using the t-Distributed Stochastic Neighbor Embedding algorithm (t-SNE) [23], which gives some insights into what should restaurants with certain attribute looks like as shown in Figure 10. It's interesting to find that the photos seem unrelated to the labels, which indicates that the hidden deep features are more effective than interpretable information from photos. This explains why the performance of HOG models are significantly worse than CNN models. What's more, the weak correlation between instance and label emphasizes the importance of utilizing deep learning methods in solving MIMLL problems.

ambience_is_classy *has_alcohol* *outdoor_seating*

Figure 10. t-SNE visualization of photos from certain types of restaurants in the test set

VI. Future Work

Considering the limitation of an individual project, this project can still be improved in several dimensions. There still exist several ways to extract features from the photos. Since the HOG-extracted features and the CNN-extracted features are of the same size in this project, weighted feature fusion is realizable and can be utilized to improve the quality of features. HOG can be improved by testing different cell sizes and block sizes. The Inception-v3 used in this project is pre-trained on the ImageNet database and it can be appropriately fine-tuned for this specific problem. Inspired by K. Simonyan and A. Zisserman's work on large-scale image recognition using very deep convolutional neural networks(DCNN) [24], digging further into the depth of convolutional network and adding nodes to the fully-connected layer would be a proper direction to improve the image classification results in the future.

As for the MI part, EMD and other algorithms introduced by J. Amores [12] can be further tested. The performance of EMD is moderate in my implementation though it's much more complicated than SMI. For the ML part, considering possible inner correlation may be helpful to improve the results. The attribute 0 (good for lunch) and attribute 1(good for dinner) might be correlated since there tends to be a preference on one of them such that few restaurants are good for both lunch and dinner in the train set.

Based on this MIMLL process, the backward filtering of photos, that is, deleting or understating the useless photos is also realizable. What's more, this competition aimed to give tags to restaurants rather than the food itself for business reasons. Food classifiers is a further interesting direction for investigation if tags of food categories are available.

VII. Conclusions

This project solves the Yelp photo classification problems with a final F_1-score of 0.8036 on 70% data of the test set, which is a fairly good result compared to the highest score of 0.8318 on Kaggle. The model using Inception network, SMI algorithm, and a combination of SVM, RF and LR classifiers gives the best result. The deep features extracted from DCNN is more effective in prediction than shallow features. EMD algorithm makes the integrated features misleading in my experiment. The bag-level information is biased since the distribution of instances in each bag is extremely unbalanced as shown in Figure 2. Other MI algorithm on the instance level might be useful to improve the results. Deep learning methods like DCNN are essential and effective in handling complicated MIMLL problems. To summarize, this project shows a comprehensive process and a convincing strategy to deal with MIMLL classification problems.

Works Cited

[1] Kaggle Inc, "Yelp Restaurant Photo Classification," Yelp Inc., 2015. [Online]. Available: https://www.kaggle.com/c/yelp-restaurant-photo-classification. [Accessed 1 October 2019].

[2] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection".

[3] D. G. Lowe, "Object recognition from local scale-invariant features," *iccv,* vol. 99, no. 2, pp. 1150-1157, 1999.

[4] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems,* pp. 1097-1105, 2012.

[5] O. Yakhnenko and V. G. Honavar, "Multi-Instance Multi-Label Learning for Image Classification with Large Vocabularies," *BMVC,* pp. 1-12, 2011.

[6] Z.-H. Zhou and M.-L. Zhang, "Multi-instance multi-label learning with application to scene classification," in *The 19th International Conference on Neural Information Processing Systems*, MIT Press, 2006.

[7] H. Hoashi, T. Joutou and K. Yanai, "Image recognition of 85 food categories by feature fusion," *2010 IEEE International Symposium on Multimedia,* pp. 296-301, 2010.

[8] Y. Kawano and K. Yanai, "Food image recognition with deep convolutional features," *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication,* pp. 589-593, 2014.

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein and others, "Imagenet large scale visual recognition challenge," *International journal of computer vision,* vol. 115, no. 3, pp. 211-252, 2015.

[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions," *Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. 1-9, 2015.

[11] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the inception architecture for computer vision," *Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. 2818-2826, 2016.

[12] J. Amores, "Multiple instance classification: Review, taxonomy and comparative study," *Artificial intelligence,* vol. 201, pp. 81-105, 2013.

[13] J. Read, B. Pfahringer, G. Holmes and E. Frank, "Classifier chains for multi-label classification," *Machine learning,* vol. 85, no. 3, p. 333, 2011.

[14] J. Read, L. Martino, P. M. Olmos and D. Luengo, "Scalable multi-output label prediction: From classifier chains to classifier trellises," *Pattern Recognition,* vol. 48, no. 6, pp. 2096-2109, 2015.

[15] D. Heider, R. Senge, W. Cheng and E. Hullermeier, "Multilabel classification for exploiting cross-resistance information in HIV-1 drug resistance prediction," *Bioinformatics,* vol. 29, no. 16, pp. 1946-1952, 2013.

[16] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart and T. Yu, "scikit-image: image processing in Python," *PeerJ,* vol. 2, p. e453, 2014.

[17] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang and Z. Zhang, "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems," *arXiv preprint arXiv:1512.01274,* 2015.

[18] Y. Ding, J. H. Sohn, M. G. Kawczynski, H. Trivedi, R. Harnish, N. W. Jenkins, D. Lituiev, T. P. Copeland, M. S. Aboian, C. Mari Aparici and others, "A Deep Learning model to predict a diagnosis of Alzheimer disease by using 18F-FDG PET of the brain," *Radiology,* vol. 290, no. 2, pp. 456-464, 2018.

[19] Y. Rubner, L. J. Guibas and C. Tomasi, "The earth mover's distance, multi-dimensional scaling, and color-based image retrieval," *Proceedings of the ARPA image understanding workshop,* vol. 661, p. 668, 1997.

[20] S. Angenent, S. Haker and A. Tannenbaum, "Minimizing flows for the Monge--Kantorovich problem," *SIAM journal on mathematical analysis,* vol. 35, no. 1, pp. 61-97, 2003.

[21] J. B. Orlin, S. A. Plotkin and E. Tardos, "Polynomial dual network simplex algorithms," *Mathematical programming,* vol. 60, no. 1-3, pp. 255-276, 1993.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg and others, "Scikit-learn: Machine learning in Python," *Journal of machine learning research,* vol. 12, pp. 2825-2830, Oct 2011.

[23] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research,* vol. 9, no. Nov, pp. 2579-2605, 2008.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.