Yelp Restaurant Photo Classification

Team: GXYZ
Member: Jiyang Ge, Xinmeng Li, Yan Li, Ziyue Dong

Abstract: The goal of this project is to build a model that automatically labels user-submitted photos with associated labels. We explored different image feature extraction techniques and classification methods to classify these images. After evaluation, we find that CNN feature extraction with SVM model is the most accurate one as for all accuracy score, F-1 score and hamming loss.

1. Business Understanding

Image recognition is attracting our attention with the development of information technology. People are more willing to receive and share information in the form of images. One common example is the restaurant review. Yelp, a top business directory service and crowd-sourced review forum, is well-known as a platform for users to share comments of business with pictures, which boosts the efficiency of the decision making about one of the most important daily questions- where to eat and what to eat.

As data grows massively, businesses such as Yelp attempt to figure out ways to penetrate layers of information and support intelligent decision making. Since the platform receives thousands of photos uploaded by users everyday, the company should take advantage of these data- to 'read' the information of the images and put it to use. However, the biggest challenge is that these photos all have different contents, that may include lunch, dinner, drink, outdoor seating spaces, etc.

Data mining might be the most reasonable approach to solve the aforementioned business problem, as it is able to offer precise classification of given images if the model is well-trained with considerable dataset of labelled images. This project will combine machine learning, deep learning and computer vision tools to classify photos automatically to provide better services.

2. Previous Works and approaches

2.1 Review of winner's works

This problem was a Kaggle Challenge by Yelp. The winning team used different photo-level feature extraction of CNN models including Inception-V3, Inception-BN and ResNet., combined with business-level feature extraction (feature pooling, fisher vectors and VLAD descriptor), and classification models including gradient boosting, logistic regression and multi-output neural network . After this process, the weighted average is used for evaluation.

2.2 Differences of our approach and winners' approach

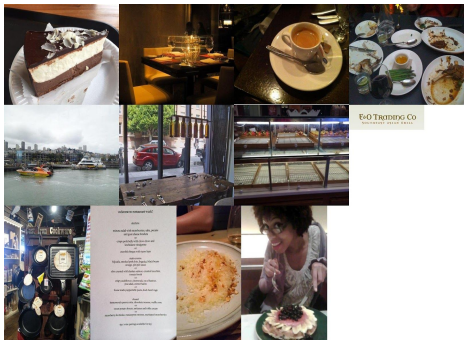
Unlike the winner's team that only used CNN feature extraction, we do vary types including Principal Component Analysis(PCA), Histogram of oriented gradients(HOG) and convolutional neural network(CNN). Moreover, we do not use the same CNN methods as the winner's, we include CNN methods of VGG16, ResNet50, ResNet152 and InceptionResNet152. Due to time constraints and knowledge limitations, we do not do the business-level feature extraction.

For the choices of models, the winner's team used five different classifications as binary relevance, ensemble of classifier chain logistic regression, multi-output neural network, binary relevance XGB and ensemble of classifier chain XGB. Unlike them, we attempt to touch simpler and more basic

models such as Logistic Regression(LR), support vector machine(SVM) and Random Forest(RF). The LR and SVM models are mentioned in the class and are also common methods in image classifications. We also choose these models based on some literature as there are several professional academic papers that introduce different image classification methods[1][2].

3. Data Understanding

3.1 Data source



The dataset was posted by Yelp's in 2015. From https://www.kaggle.com/c/yelp-restaurant-photo-classication.

The left side is some sample images from the dataset.

3.2 Describe of data

The given training dataset provides the labels for 2,000 restaurants, 234,842 photos, a map from each business ID with its associated correct/truth labels, and a mapping from each photo to its associated business ID. The nine labels including 'good_for_lunch', 'good_for_dinner', 'takes_reservations', 'outdoor_seating', 'restaurant_is_expensive', 'has_alcohol', 'has_table_service', 'ambience_is_classy', and 'good_for_kids'. Also, as the number of photo-restaurant pairs and the number of restaurant-label pairs in the given file are both higher than the numbers of photos. This means, it allows each business to have more than one picture, and allows each picture to have more than one picture.

For the given test dataset, there are 237,152 photos and 10,000 restaurants. However, as the outcomes of the test dataset are not released, we use only the training dataset and split this dataset into training and testing set.

4. Data Preparation

Firstly, we relate the images with associated labels by matching the paris of business ID and photo IDs. As we ignore the business ID, we will lose information about how the pictures are grouped by the same restaurant and background. However, this could be useful for prediction, since images of the same business may be related.

In order to run the models appropriately, we transform the multilabel classification into 9 independent single label classifications by assuming each label is independent and use one vs rest(OvR) strategy on 9 binary classifiers.

Due to the GPU and RAM constraints, we randomly choose 20000 photos from the dataset and split it to training data of 16000 photos and test data 4000 photos. Also, as the sizes of images

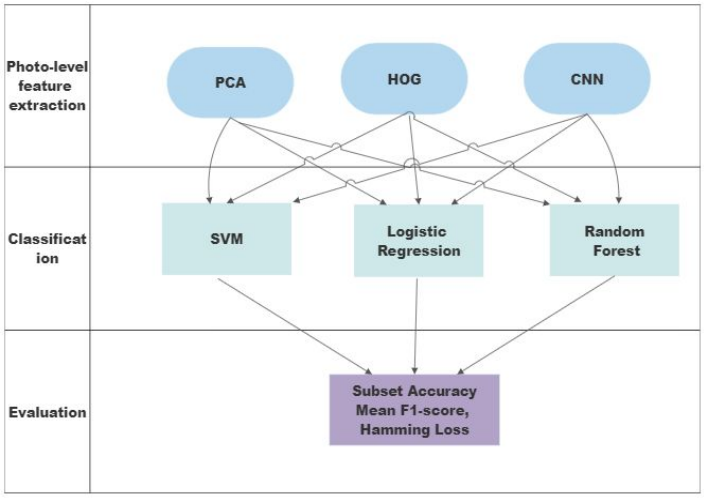
varies a lot, in order to unify the formats for applying models and exploring features, we resize all images to 256*256 pixels and transformed them into numpy arrays of size(256,256,3).

After one-hot encoding the categorical target labels, we have a total of 9 target variables as Y_0, Y_1, …, Y_8. Each Y_i is a binary random variable indicating whether the associated image is labeled as 0: good_for_lunch, 1: good_for_dinner, …, 8: good_for_kids or not.

Since all target variables are binary, it does not make much sense to find the value of minimal, 25% percentile, 75% percentile and maximum of each label set. The mean and standard value of each label is computed as follows:

	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8
mean	0.228043	0.594259	0.642751	0.488439	0.422139	0.737530	0.770088	0.371846	0.506311
std	0.419571	0.491036	0.479190	0.499867	0.493902	0.439978	0.420777	0.483299	0.499961

5. Feature Extraction and Models



Our input dimension is (256*256*3), which is too high, as a result, the learning algorithm would be extremely slow. Photo-level feature extraction techniques are used to reduce the desired memory space and speed up the fitting of machine learning algorithms. We try three different ways of feature extraction including Principal Component Analysis(PCA), Histogram of oriented gradients(HOG) and convolutional neural network(CNN). Then we use the compressed dataset to apply different machine learning models including Logistic Regression(LR), support vector machine(SVM) and Random Forest(RF).

5.1 Data Mining Algorithms

Before talking about each feature extraction technique, we compare our three classification models horizontally.

5.1.1 Logistic Regression

Logistic Regression is one of the most widely used algorithms in binary classification. It runs fast, and we can use it as a baseline. However, it assumes the data to be linearly separable, so we deduce that it only works well when there is a small portion of categorical variables in features.

5.1.2 SVM

Compared to logistic regression, SVM performs well on classification no matter if the data is linear or not. However, nonlinear kernels such as RBF, is usually not scalable due to its memory intensity.

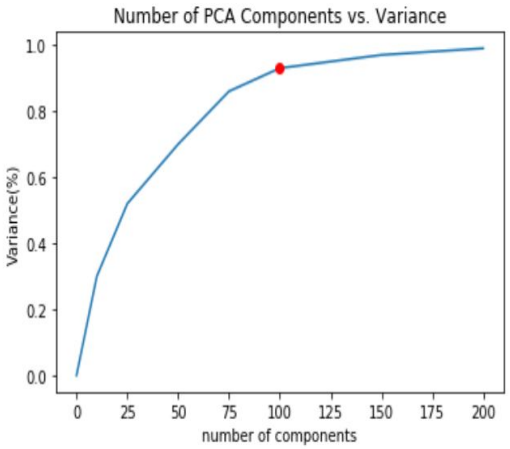
5.1.3 Random Forest

The random forest model reduces the probability of overfitting by averaging the prediction results of a diverse set of trees. It is always compared with the decision tree - the total bias of random forest is higher, as it trains independent trees on random samples of data. However, the final prediction result is the average of all individual trees, which alleviate the increased bias by reducing variance. Therefore, the accuracy of the random forest is always slightly higher than the decision tree.

5.2 Feature extraction via PCA

The most common method to speed up machine learning algorithms is to use Principal Component Analysis (PCA). PCA is a powerful technique used for dimensionality reduction - it directly decreases the number of feature variables and saves on computations, but still retains the most valuable parts of all of the variables, which keeps the most important information of the image. Also, the pca-processed data are all independent of one another. This is a benefit because the assumptions of a linear model require variables to be independent to one another.

5.2.0 PCA component choice



The number of principal components to retain in a feature set depends on several conditions such as storage capacity, training time, performance, etc. In order to find the optimal component number, we take the number of principal components that contribute to significant variance and ignore those with diminishing variance returns. Specifically, we do experiments with components n=[5, 10, 25, 50, 75, 100, 150, 200]. Considering the balance between training time and performance, we choose to reduce the 256*256*3 dimensional data into 100 dimensions, as it captures 92.86% information of the given image and does not take too much RAM space when transforming the data at the same time.

5.2.1 pca+ lr (Non-standardized vs. standardized)

The baseline model is the default logistic regression model. This model gives subset accuracy score of 0.607, f-1 score of 0.509 and hamming loss of 0.393. We usd GridSearchCV to find better parameters and do cross validation to improve this model. The best result can be achieved with setting L1 penalty, c=1 and liblinear as a solver. With this setting, the model gives an accuracy score of 0.608, f-1 score of 0.511 and hamming loss of 0.392. Both accuracy and f-1 have been improved a little bit but very insignificant.

Another way we work on to improve the performance of the baseline model is to standardize the output variables before applying PCA. We transform the data onto unit scale as mean is zero and variance is one, so that the scale of features have less effect on PCA. StandardScaler is used to

standardize the dataset's features by fitting on the training set and transforming on the training and test set. Also, we choose the minimum number of principal components such that 92% of the variance is retained.

Then for each binary classifier, we apply the default Logistic Regression model for sklearn and generate a likelihood for each label. As if the probability is greater than or equal to 0.5, then we predict the as 1. Otherwise, we predict as 0. In order to increase the calculation and running speed, we change the default solver to 'lbfgs'.

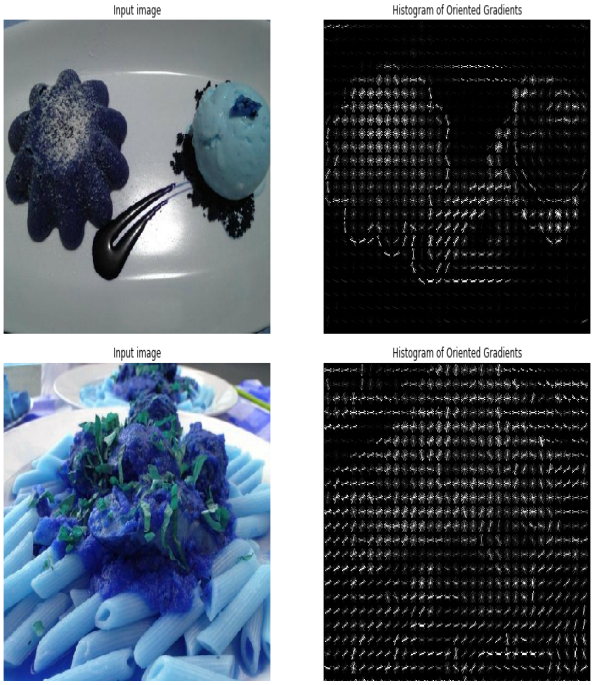
5.2.2 PCA+SVM

For the Support Vector Machine(SVM) model, the performance of the best model we can achieve is the default model. GridSearch is used to find the optimal parameters, but due to the nature of the SVM algorithm, the running time is extremely high, we consider this combination as an inefficient approach.

5.2.3 PCA + Random Forest

For the Random Forest model, we use GridSearchCV in sklearn as cross-validation to get the result with the associated parameters that has the greatest accuracy among all given optimal ranges of parameters. We test several choices of parameters including max_features in range(1,11), min_samples_leaf in range(1,3), max_depth in range(3,13), and criterion in range ['gini','entropy'], as these features are important features of decision tree. The best result comes with kernel = rbf and gamma = 0.06. After comparing the accuracy Score, recall Score (how much of malignant tumours were predicted correctly) and precision Score (how much of tumours, which were predicted as 'malignant', were actually 'malignant'), we get the best parameter combination is that max_features = 4, min_samples_leaf = 1, and max_depth=7.

5.3 Feature extraction via HOG



While PCA reduces needed computing power by dimension reduction, Histogram of oriented gradients(HOG) removes unnecessary information, and detects edges in a photo, which can incredibly decrease the file size. After applying HOG, we got a 150 megabytes(20000 images) ".h5" file, which stores directed vectors.

In our project, first we change previous colorful images to black and write, since HOG is not color sensitive.It calculates the image gradient of each pixel horizontally(x axis) and vertically(y axis). Value of each pixel is in a range of 0(black) and 255(white). All vector gradient angel of $atan(\frac{\Delta y}{\Delta x})$ and gradient magnitude of $\sqrt{\Delta x^2 + \Delta y^2}$.

In our feature extraction process, we used 16*16 pixel cell to compute gradient magnitude and direction, and split 360° to eight 45°.

Right hand side shows feature vectors after HOG. We can see edges clearly in upper right image, however, HOG does not do a good job on the lower left image.It is reasonable, because HOG only extract surface features, when there are too many objects in one image or there are too many noises in the background, HOG would not perform well compared to a deep learning methods like ResNet.

After extracting feature vectors using HOG, we try 3 different data mining algorithms: logistic regression, SVM, random forest. As a baseline, we run these three models with default parameters. To improve our model, we apply grid search and cross validation(5 fold) to logistic regression. The best parameters combination is: penalty = 'l2'(choice of 'l1' and 'l2'), and C = 0.01(among 0.01, 0.1, 1, 10). For SVM, we find kernel = 'linear' is better (compared to kernel = 'rbf'). We use grid search for parameters for min_samples_leaf, and we find min_samples_leaf = 35 gives the best result.

5.4 feature extraction via CNN

Since it is time consuming for us to build a neural network from scratch for image processing, we use the deep learning model with pre-trained weights at first to extract features and then implement multi-label-multi-class classification on it.

5.4.1 Pre-trained Model Selection

First, we select candidates for the optimal pre-trained model. Based on the information of Keras' available pre-trained models on ImageNet, we observe that InceptionResNet152 and NASNetLarge have the highest validation accuracies. Since the validation accuracies of these two models are close and NASNetLarge requires larger input size (331x331x3) than InceptionResNet152 (299x299x3), we choose InceptionResNet152 as a candidate in consideration of our memory limit. Since VGG and ResNet are widely used deep networks for large-scale image recognition, we also select VGG16, ResNet50 and ResNet152 as our candidates. The input size of all of these three models is 224x224x3.

Next, we preprocess our original images and use these selected models to extract features with Keras. The output feature matrix is then split into training and testing dataset. To ensure consistency, we compare the accuracy of the results of running logistic regression models in a default setting for all of these four pre-trained models. The test accuracy for VGG16, ResNet50, ResNet152 and InceptionResNet152 is 0.6590, 0.6776, 0.6924, 0.6916 respectively. Based on the comparison of accuracies and the size of input, we finally choose to use ResNet152 as our feature extraction model.

5.4.2 Model Parameter Optimization

After extracting informative features from images, we leverage machine learning models to classify the labels associated with each photo. In this stage, parameter tuning is essential to find the optimal model with an excellent performance. During the grid search for optimal parameter values, we use 3-fold cross validation, i.e. train each parameter combination three times, AUC scoring, i.e. estimator with the highest AUC score will be chosen, and set n_job = -1 such that we can save time by running it in parallel. To reduce the bias from GridSearch in the final comparison of SVM,

Logistic Regression and Random Forest, we keep the parameter setting of GridSearch consistent across these three types of models.

5.4.2.1 SVM

Since the Support Vector Classifier is not computationally efficient, we only implement grid search on LinearSVC(the adapted SVM model for large-scale dataset). We try to run SVM on the default setting with an rbf kernel, which assumes the boundaries to be curve-shaped. It turns out that it is time-consuming to run SVM on a large dataset with a non-linear kernel even for only one parameter combination.

We iterate through five C values from 0.001 to 10, and find out that SVM has the best performance with C=0.001, which implies that SVM prefers weaker strength of regularization. The AUC score of the best parameter is 0.693.

5.4.2.2 Logistic Regression

Since our dataset is large and it might be slower for the solver function to converge. We increment the limit of the maximum number of iterations to 50000.

Since we train the model for each target variable, we get 9 optimal parameter combinations from grid search. For each parameter, we want to find out a value such that the model is capable of performing reliably on as many classes as possible. To deal with the variation of parameter settings, we choose the value that appears most from all of the possible optimal options of the best estimators of 9 classes. Based on the results below, we choose C=0.1, solver = saga as the top combination. Newton-cg is not chosen due to its slow computation.

Let # represents the number of classes that has this parameter value in their best estimators. Then # = 8,1,0 respectively when C=0.1,1 and 10. # = 3,1,2,3 when solver = newton-cg, lbfgs, liblinear, and saga.

5.4.2.3 Random Forest

Based on the results, we choose max_depth=None, n_estimators = 300, and min_sample_split = 2 as the final combination. We get # = 0,1,8 respectively when max_depth = 5, 10 and no limit. # = 4, 5 when the number of estimators is 200, 300. # =5,4 when min_sample_split = 2,5. We observe that the votes for two parameter values are close.

6. Evaluation

6.1 Evaluation metrics

In order to compare the multilabel classification models, we use several multilabel evaluation metrics to measure the accuracy of results predicted by the models.

6.1.1 Subset Accuracy

For any classification problem, the most basic evaluation metric is the accuracy scores as mean prediction accuracy. The accuracy is calculated as Accuracy = Prediction / Actual. Since this problem is mutilabelled, we use the subset accuracy, which is the average of the accuracy value of

each label. Intuitively, it means in what percentage the images can be labelled in any label correctly on average.

6.1.2 Mean F1 score

The evaluation metric of F1-Score is well-known as an example-based F-measure in multi-label learning literature. The F1-Score measures the accuracy based on precision(P) and recall(R). The F1-Score is calculated as following:

$f1 = 2 * \frac{Recall * Precision}{Recall + Precision}$ where $precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$

In this case, we use the Mean F1 Score, the average of the F-1 score of prediction of each label, to evaluate our model. Models with higher Mean F1 Score are considered better.

6.1.3 Hamming Loss

Hamming loss is the evaluation metric used for multi-label classification in sklearn. It is the fraction of labels that are incorrectly predicted, as it is calculated the number of wrong labels over the number of total labels. In this case, model with minimal Hamming loss is considered as better performed

$$\frac{1}{|N| \cdot |L|} \sum_{i=1}^{|N|} \sum_{j=1}^{|L|} \text{xor}(y_{i,j}, z_{i,j}),$$ where $y_{i,j}$ is the target and $z_{i,j}$ is the prediction.

6.2 Comparison of All Models

The following is a summary of scores for all methods measured by different evaluation metrics:

Feature Extraction	Classification Model	Subset Accuracy	Mean F1 score	Hamming Loss
PCA	Non-standardized LR**	0.607	0.509	0.393
	Standardized LR	0.643	0.598	0.375
	SVM	0.631	0.431	0.369
	Random Forest	0.672	0.618	0.256
HOG	Logistic Regression	0.651	0.552	0.348
	SVM	0.592	0.561	0.408
	Random Forest	0.634	0.534	0.359
CNN	Logistic Regression	0.717	0.688	0.283
	SVM	0.743	0.704	0.257
	Random Forest	0.725	0.641	0.275

**The Non-standardized Logistic regression model with PCA is the Baseline Model.
*The top two performed evaluation results are highlighted.

From the above table, we can see that CNN generally performs better than the other two, and HOG performs similar to PCA. Among the classification methods, the differences between each method perform are relatively small and depend on the choices of feature extraction. Among all the

combinations of feature extraction and classification models, CNN with SVM model is the one with the generally highest accuracy.

6.3 Evaluation of The Chosen Model

As the chosen model has the average accuracy of over 0.7, it could have a relatively high probability to predict the associated labels correctly. Thus, the business is able to automatically tag the users' photos uploaded directly through the model instead of labeling millions of images by human beings. This is not only time-efficient but also saves money and labor. Therefore, the business could put this amount of reduced cost into other fields and then make more profits. Also, customers could get read-time information as they can see images uploaded hours ago and have a better experience.

7. Deployment

7.1 Deploying the Result

In the real world, our predictive model could be widely deployed to help a variety of groups of customers automate their decision process. We discuss two well-defined potential target markets from software companies and application users from three dimensions including motivation, benefit and cost.

7.1.1 For software cooperation

The primary goal of corporations and businesses is to generate profits. Our model aims at helping software development firms in the catering industry acquire more profitable customers through "better targeting, reduce the cost of acquisition in the early stages of the relationship, cross-sell and upsell to the right customers and increase wallet share through loyalty, retention and recovery program"(Forrester)[5]. Our model makes it more convenient and smoother for existing software users to discover more restaurants that match their tastes and needs. A user who is curious about a specific restaurant is able to acknowledge the characteristics of a restaurant by taking a look at the automated-classified labels such as taking reservations in a few seconds. A user who would like to find a restaurant also feels satisfied if the application enables him or her to click a label and skim through the associated photos. Moreover, the software company can incorporate our photo classification model into a recommender system to provide personalized recommendation. For example, it is very likely that a user who likes and visits kid-friendly restaurants frequently gets interested in an entertaining kid themed restaurant photo.

A recent research by American Express indicates that 60% of customers are willing to pay more for a better experience. Through the above ways, our model improves the customer satisfaction rate through the photo classification feature, which not only prevents the existing users from churn and elevates the loyalty, but also attracts new users and makes users who in an early stage form a habit of using the software faster. Our model is also affordable, we can just wrap it and deploy it to cloud.

7.1.2 For application user

There are two groups of people using software companies like Yelp, business owners such as restaurant managers and consumers who are looking for services. The primary goal of a business owner, similar to software cooperation, is to make a profit. Our model provides an opportunity to

them to arrest the attention of new consumers, especially for older adults restaurant owners who are not familiar with smartphones. Consequently, little-known restaurants with delicious food and great services would appear on the searches and the recommendations associated with the label and could be discovered faster. The customer wants to find out an ideal service provider precisely and fast on the platform of the application.

Obviously, our model is helpful in this processing of matching service providers with their potential consumers. It also costs nothing for both retailers and customers to use this photo classification function integrated in the application.

7.2 Consideration for Firms

7.2.1 General Issues

The accuracy of the best model is around 0.74, and therefore there must exist misclassified images, which could be misleading and disappointing for application users. We suggest the software companies add a report button on the searching or recommending page. Once the user clicks it, a wrong classification result would be reported, which could be added to the model as new training instances. As a result, the model would be adjusted to the company's data. Also, our model is only trained on a sample of the original dataset due to the computation power limit. Therefore, it is not very scalable and it is very likely to perform worse when we apply it into a real-world software with millions of images to be processed in the database.

7.2.2 Ethic Consideration

As shown in the samples of the dataset, images with people's faces are also contained . Will those images be securely stored and won't be used for other purposes? Does yelp own these data? Do they have the right to sell data to other data analysis companies? All those questions are not clear and people need to think about it and be aware to protect their information security.

7.2.3 Risks and how to Mitigate

The privacy disclosure is considered as the worst scandal for companies like Yelp. Shared private information should be treated confidentially. Third party companies that share data like financial or locational need to have restrictions on whether and how that information can be shared further. Also,customers should have a transparent view of how our data is being used, and the ability to manage the flow of their private information across massive, third-party analytical systems.

8. Future Work and Limitation

The considerations of this project are time and computation limitations including RAM, DISK, GPU and etc. Even though we used only 20,000 images, several models run for hours, for example, PCA and Random Forest model runs for 2 hours for each label. Thus, while dealing with large dataset, it could be time-consuming and require high-quality computational equipment. Also, one problem with our evaluation method is that there's a chance of being partially correct, but all of our evaluation methods only focus on the overall matches and ignore the particle correct matches.

Reference

[1] Yunpeng Li, David Crandall and Daniel Huttenlocher. Landmark Classification in Largescale Image Collections. 2009, IEEE. 1957-1964.

[2] Anna Bosch, Andrew Zisserman and Xavier Munoz. Representing shape with a spatial pyramid kernel. 6th ACM international conference on Image and video retrieval. 401-408.

[3] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, San Diego, CA, USA.

[4] https://www.learnopencv.com/image-recognition-and-object-detection-part1/

[5]https://www.forrester.com/report/How+Analytics+Drives+Customer+LifeCycle+Management/-/E-RES60149

Appendix (Members contribution)

The whole group do the research and extract study together, and then split responsibilities for each group member:

Ziyue Dong: Optimal component of PCA, Feature exploration PCA with baseline and svm model

Jiyang Ge: Feature exploration PCA with standardized logistic regression and random forest

Yan Li: Feature exploration HOG with three models

Xinmeng Li: Feature exploration CNN with three models