COMP9517 2020T3 PROJECT

Individual component

Xiaoyu Dong

z5323011@ad.undsw.edu

***Abstract*—Image-based plant phenotyping is a growing application area of computer vision in agriculture. [1] The aim of this paper is to assessing the bag of word(BoW) solution and comparing several models on this classification task. The BoW is the most popular method for feature encoding. BoW encodes the image features into a fixed- dimensional histogram for representing the image.**

The experimental results shows the performances using the mean f1-score and ROC AUC criteria.

Keywords—segmentation, visual bag of words feature; watershed; SURF; SIFT; kmeans; vector quantization; image classification

I. INTRODUCTION AND BACKGROUND

Image classification is an important part of computer vision subject, which can be used for image and video retrieval, digital library management.

This task is to distinguish Arabidopsis plant rgb images from tobacco plant rgb images from the Ara2013-Canon and Tobacco folders of the Plant Phenotyping Dataset.

To complete the image classification in this task, watershed is applied for segmentation; then the procedure of BoW includes using SURF to extract the set of local keypoint descriptors from the training image data; codebook generation; using K-means clustering and vector quantization to generate the bag-of-words feature matrix(creating the vocabulary, which represents the categories of local descriptors); Finally, by comparing the performance metrtics of 4 models in machine learning - SVM, Random Forest, Logistic Regression, K Neighbors Classifier to get the best solution for this task.

II.METHOD (IMPLEMENTATION)

A. Watershed

Any grayscale image can be viewed as a topographic surface where high intensity denotes peaks and hills while low intensity denotes valleys. The "philosophy" behind the watershed is filling every isolated valleys (local minima) with different colored water (labels). Building barriers in the locations where water until all the peaks are under water. The barriers created is the segmentation result.

B. Using the Scale Invariant Feature Transform (SIFT) or SURF to detect the feature and create the vocabulary

SURF and SIFT are both good approach for detecting the keypoints of images. SIFT has a better effect in the case of different scale images and rotation transformation while SURF has a better matching effect when illumination change. The plant has different size and are at different orientation. The performance of these two method will be compared in this task.

There are 4 procedures in SIFT: Scale-space extrema detection, Keypoint localization, Orientation assignment, Generating keypoint description. After constructing a SIFT object, finding keypoints and descriptors.

In order to create the vocabulary of Bow, which represents the categories of local descriptors, lists of each descriptors of images need to be made.

C. codebook generation(kmeans clustering(scipy.cluster.vq.kmeans)

k-means algorithm adjusts the classification of the observations into clusters and updates the cluster centroids until the position of the centroids is stable over successive iterations. In this implementation of the algorithm, the stability of the centroids is determined by comparing the absolute value of the change in the average Euclidean distance between the observations and their corresponding centroids against a threshold. [2], k-means is one of the most popular unsupervised learning method, to partition the training data into multiple categories. A centroid is a vector that contains a number for each variable, and each number is the mean of a variable for the observations in that cluster. Scipy.cluster.vq.kmeans returns the codebook, which is a k by N array of k centroids(i.e. the length of codebook equals to k). The i-th centroid codebook[i] is represented with the code i. K-means algorithm mapping centroids to codes and vice versa. [2]

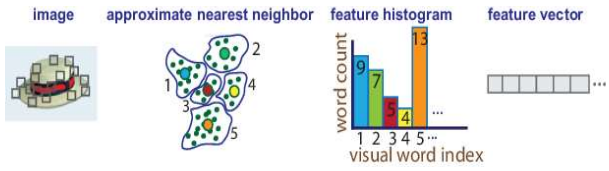

Fig.1

D. *creating a matrix of feature*

Creating a matrix, each row of this matrix should represents each images in the applied dataset, and each column represents each centroid as shown in Fig.1.

E. *vector quantization(scipy.cluster.vq.vq) and updating the matrix of feature*

This approach assigns a code from the code book created in part B to each observation. Each observation vector in the 'M' by 'N' obs array is compared with the centroids in the code book and assigned the code of the closest centroid. In the base of the proximity of the descriptor to a particular cluster center, incrementing the histogram bins. Therefore, the histogram length equals to the number of visual words that the Bag Of Words constructed and finally the histogram bin is able to represents the image.

The features in obs should have unit variance, which can be achieved by passing them through the whiten function. [3]

F. *k-fold cross-validation*

It is used to optimize the model parameters(e.g. n_neoghouber of KNeighbors Classifier). The training set is split into k parts. Training happens k times, each time leaving out a different part of the training.

As a result, the performance of the applied model can be validated on multiple folds of data, therefore, k- fold cross validation have an advantage if there is a unbalanced distribution of the predicted object's classes.

This is also used in the performance comparison to get a mean AUC ROC of 10 folds in Ⅳ. RESULTS AND DISCUSSION for assessing the performance.

G. *Fit the model and evaluate the metrics*

Four models are trained and tested in this task.

- Support Vector Machine: it constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression, or other tasks like outliers detection.[4]
- Random Forest Classifier: it constructs numerous individual decision trees at training time and outputs the classification or regression of the individual trees. The class with the maximum votes becomes the model prediction. Many uncorrelated models operating together, therefore, generally, it has a better performance as it corrects for decision trees' habit of overfitting to their training set.
- Logistic Regression Classifier: It is a statistical model used to predict the categorical, it constructs a logistic function to model a binary dependent variable.
- K Neighbors Classifier: A classifier based on the k-nearest samples.

H. *Metrics(f1-scores, AUC ROC)*

F1-score is the weighted average of Precision and Recall. Which means it takes Precision and Recall into consideration at the same time. Therefore, this score takes both false positives and false negatives into consideration. F1-score is more useful than accuracy in the case of class-imbalance distribution.

ROC (Receiver Operating Characteristics) curve is the relationship between True Positive Rate and False Positive Rate given by different threshold. AUC ROC is the area under the ROC curve. It equals to the probability that a randomly selected positive example has a higher risk score than that of a randomly selected negative example. Therefore it is used to measure how well the model can distinguish different classes. The bigger of the area under the roc curve, the closer the curve to the top left corner, i.e. true positive rate=1, false positive rate=0, which means the model is more ideal for the classification task. So, It is a measure of separation between score distributions, and it does not necessarily have relationship with classification. That is the reason why it is suitable for the case of imbalance-class distribution.

III. Experiment

A. *Read the images and add label to different class of plant*

Using os.listdir to read each image and resize them to a exactly same size. If the plant is Arabidopsis, the label is 0, if the plant is Tobacco, the label is 1. Creating two lists, one contains all image matrix which is the X and another one contain all labels of each image.

- Add image lists of Arabidopsis and Tobacco together as the X dataset.
- Add label lists of Arabidopsis and Tobacco together as the y datasets.

B. *Split dataset into training and test dataset*

Shuffles all datasets, pick some parts as the training data, and other part as the test dataset.using shuffle method give a more balanced performance. 80% of the whole dataset is the training and validation set, and 20% is for the test.

C. *watershed*

Segment the image, then identify the amount of each pixel value of the image, process the RGB image's background to black.

D. *Using SIFT or SURF to detect the feature and create the vocabulary*

After processing the image with watershed, SIFT is not suitable for this task, as the parameter - nfeature cannot guarantee there is descriptor can be detected from every image. So SURF is selected as the tool. Fig.2 shows the keypoints detected.

In order to create the vocabulary, which represents the categories of local descriptors, lists of each descriptors of images need to be made.

E. *Codebook generation(kmeans clustering(scipy.cluster.vq.kmeans)*

In this task, the number of centroids is set as 130 to generate the codebook, Fig.3 shows the whitened data in blue and cluster centers in red. The number of centroids should not be set too small incase no enough column to represent the different features detected in each image, which result in bad performance.

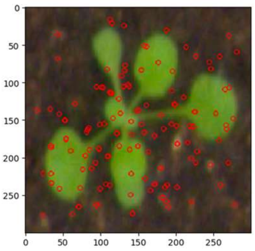

Fig.2

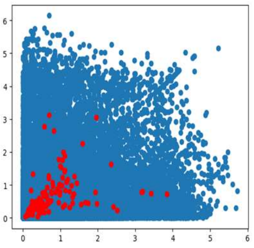

Fig.3

F. *Creating a matrix of feature*

Creating a zero matrix, whose value of each index will be updated later. The height of this matrix is the number of all images in the applied dataset. And the width of the matrix is the number of centroids generated in part B, i.e. the length of codebook.

There are several keypoints in the image, so the code, which is the return of scipy.cluster.vq.vq, some centroids index appear multiple times. It is a list like: [39 52 97 52 52 97 52 … …9 52 52 52 54… …61 86]. According to this code, updating the matrix of feature. For each image, once the index of centroid appears one time in the code list, add 1 to the index to make the histogram, which represents the image .

G. *train and test the model and evaluate the metrics*

After training and test, 10-folder cross validation method is used to give the mean scores of 10 folds to evaluate the performance of each model.

IV. Results and Discussion

In some cases, threshold of SURF should not be set too small in case too many unnecessary keypoints are detected and therefore result in a bad result. But in this task the threshold should also not be set too large in case there is no keypoints detected from the image, as the whole background has been black, so this situation may happen. By comparing the performance of setting different threshold, the performance has no big difference. Fig.4 shows the confusion matrix of SVM with SURF detecting the features, all images has been correctly classified, other models have similar performance. The two classes of images are quite different from each other, the AUC are both reach almost 1. The Arabidopsis is a big separated plant and tobacco is a tiny plant in a pot, and the background of Tobacco contains other things, which can be a different feature from the Arabidopsis. It is enough to distinguish them using bag of word method, even though without watershed to pre-process the images.

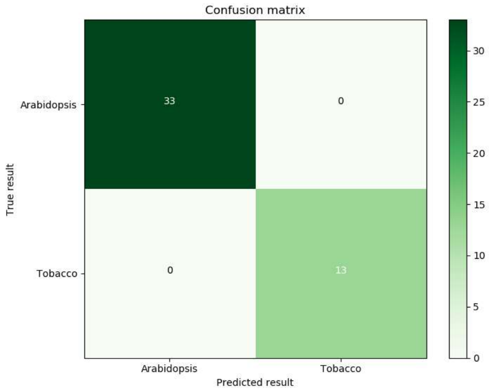

Fig.4

Fig.5 shows the ROC curve of each classifiers when the threshold of SURF is set as 399. Almost all AUC ROC are 1, which means all classifier are ideal for this task (True Positive Rate is 1 and False Positive Rate is 0, which means it correctly classify all positives and negatives).

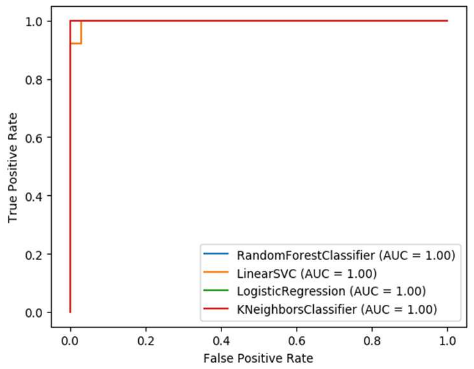

Fig.5

Metrics(mean of 10 folds)	SVM	Random Forest	LogisticRegression	KNeighbors (n=1)
F1_score	1.0	0.87	1	0.99
ROC_AUC	1.0	0.99	1	0.99

All models complete a good performance, if considering the mean f1_score , BoW method with LogisticRegression classifier and SVM result in a satisfying and stable performance for this task. But it is just very small difference.

References

[1] H. Scharr et al., "Leaf segmentation in plant phenotyping: a collation study," Machine Vision and Applications, vol. 27, no. 4, pp. 585-606, 2016/05/01 2016, doi: 10.1007/s00138-015-0737-3.

[2] T. S. community. "scipy.cluster.vq.kmeans." The SciPy community. https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.vq.kmeans.html (accessed October 28,2020).

[3] T. S. community. "scipy.cluster.vq.vq." The SciPy community. https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.vq.vq.html (accessed October 28,2020).

[4] W. contributors. "Support vector machine." Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Support_vector_machine (accessed October. 28, 2020)