Individual Component of COMP9517 Project

Liqi Jiang z5253065

***Abstract*—This paper aims to implement a python solution to perform image classification of plants into two plant classes, namely Arabidopsis and tobacco. I use SIFT algorithm to obtain visual vocabulary from the set of keypoint descriptors, then create the bag-of-words using the visual vocabulary by the means of K-means clustering. I calculate the feature vector by bag-of-words. SVM and KNN algorithm are used to recognize patterns in test images and classify them into appropriate classes based on these patterns. Finally, I got high accuracy in distinguishing the two plants.**

***Keywords*—plant, feature extraction, image classification**

I. INTRODUCTION AND BACKGROUND

Image classification is a fundamental task that attempts to comprehend an entire image as a whole. The goal is to classify the image by assigning it a specific label. Typically, image classification refers to images in which only one object appears and is analyzed. In contrast, object detection involves both classification and localization tasks, and is used to analyze more realistic cases in which multiple objects may exist in an image [1].

In terms of the goal of the individual component, it aims to use traditional feature extraction techniques and pattern recognition to perform image classification of plants into two plant classes, namely Arabidopsis and Tobacco.

The dataset used for this individual component is images whose filenames end with "_rgb.png" in the section "Plant" of Plant Phenotyping Dataset. For each image, it has its own features and the corresponding label which is represented by its belonging folder name. There are totally 227 images representing the two plants. In Figure1, subfigure (a) shows the plant Arabidopsis and subfigure (b) shows the plant Tobacco.

(a) (b)

Fig. 1. Example of Arabidopsis and Tobacco

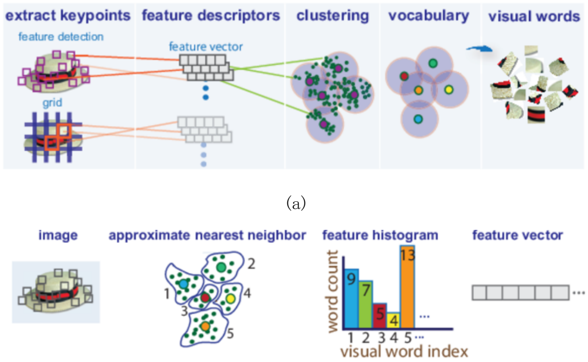
II. METHOD

A. Datapreprocessing

The sizes of images in the dataset are inconsistent. Some of them are a little bit large, which will lead to long running time. I applied cv2.resize() to set the width of each image to 100 with the same scale.

B. Image Representation

Model Bag-of-Words(BoW) is considered to be applied to image representation. In this model, we can think of an image as a document or a collection of several visual terms and these terms are not in order with each other. Since words in images are not as readily available as those in text documents, we need to first extract features from images. Figure 2 [3] shows the steps of BoW. The features of all the images are put together to form a dictionary, and then the features in this dictionary are grouped into K word bags. For each feature in each picture, find the center point nearest to the feature, that is, find the frequency of the occurrence of K word bags in the image, and use the frequency of the occurrence of these K word bags to form the feature vector of the picture.



(a)

(b)

Fig. 2. Bad-of-Words(BoW) [3]

a) Feature Extraction

Feature extraction will reduce the number of resources required to describe a large set of data [2]. SIFT algorithm is the most widely used algorithm to extract local invariant features in the images, so I used the SIFT algorithm to extract invariant feature points from the images as a visual vocabulary. It firstly builds DOG scale space. Secondly, it searches and locates the keypoints. Thirdly, the feature points should be assigned according to the local image structure of the detected key points to achieve rotation invariance. Then, keypoint descriptors can be generated. The key point descriptor includes not only the key point, but also the pixels around the key point that contribute to it. Function detectAndCompute() of SIFT will return keypoints and descriptors. Figure 3 shows a Tobacco image with keypoints. Figure 4 shows the feature match between two Tobacco images using SIFT.

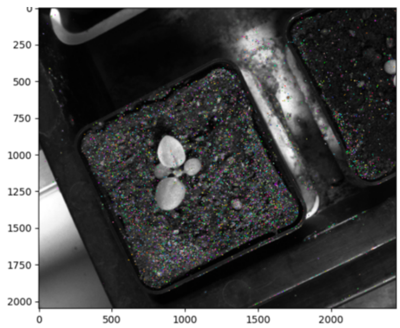


Fig. 3. a Tobacco image with keypoints

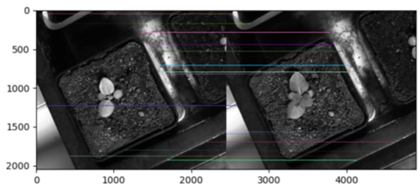


Fig. 4. feature match between two Tobacco images

b) K-means Clustering

K-means algorithm is an indirect clustering method based on the similarity measure between samples. This algorithm takes K as the parameter and divides N objects into K clusters, so that there is a high similarity within the cluster and a low similarity between clusters. SIFT extracted visual word vectors according to the distance, and k-means algorithm can be used to combine words with similar meanings as the basic words in the vocabulary. Figure 5 shows the process of building the vocabulary.

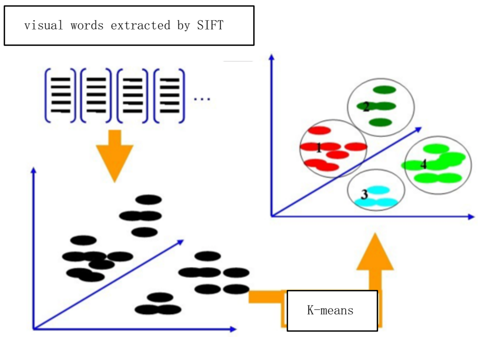


Fig. 5. construct the vocabulary using K-means

c) Feature Representation

I use the words in the vocabulary to represent images, which means calculate the feature vector of images in other words. SIFT algorithm can be used to extract many feature points from each image, and these feature points can be replaced by words in the vocabulary approximately. By counting the number of times each word in the vocabulary appears in the image, the image can be represented as a K-dimensional numerical vector.

C. Pattern Recognition

Pattern recognition is the classification of data based on already existing knowledge or on statistical information extracted from patterns and/or their representations. In this part, I tried with both SVM and KNN.

a) SVM

Support Vector Machines (SVM) is a dichotomous model. Its basic model is the linear classifier with the largest interval defined in the feature space, which is different from perceptron.[4] SVM also includes kernel tricks, which make it essentially a nonlinear classifier. The learning strategy of SVM is interval maximization, which can be formalized as a problem to solve convex quadratic programming, which is also equivalent to the regularized hinge loss function minimization problem.

b) KNN

The idea of K-Nearest Neighbors(KNN) is that a sample is most similar to k samples in the data set, and if the majority of k samples belong to a certain category, the sample also belongs to this category. It's important to find the best number for the number of neighbors used in KNN.

III. EXPERIMENTAL SETUP

I set label = 0 to class Arabidopsis and label = 1 to class Tobacco. Because there are strikingly large differences in the image sets used, I set a very small portion 20% of the whole dataset to be the training set. And when building the Bag-of-Words, I just used the training set, without the test set.

I used the accuracy, recall and precision of classification report to evaluate the performance .

$$\text{accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$\text{precision} = \frac{TP}{TP+FP}$$

$$\text{recall} = \frac{TP}{TP+FN}$$

a) Parameters of SIFT

When creating a SIFT object, there are 5 parameters that can be changed. NFeatures specifies the number of feature points to be returned and does not affect the result of SIFT feature detection. The parameters nOctaveLayers and Sigma mainly affect the composition of image Gaussian pyramid, ContrastThreshold and edgeThreshold affect the process and result of searching for extreme points in DOG. I tried to set nOctaveLayers = 3, ContrastThreshold = 0.08, edgeThreshold = 8, Sigma = 1.0, to compare with the default value.

	precision	recall	f1-score	support
0.0	0.82	0.95	0.88	132
1.0	0.77	0.46	0.57	50
accuracy			0.81	182
macro avg	0.79	0.70	0.73	182
weighted avg	0.81	0.81	0.80	182

Fig. 6. SIFT_create(0, 3, 0.08, 8, 1.0)

	precision	recall	f1-score	support
0.0	0.90	0.97	0.93	132
1.0	0.90	0.70	0.79	50
accuracy			0.90	182
macro avg	0.90	0.83	0.86	182
weighted avg	0.90	0.90	0.89	182

Fig. 7. SIFT_create(0, 3, 0.04, 10, 1.6)

In terms of accuracy, precision and recall, parameters(0, 3, 0.04, 10, 1.6) have better effect.

b) Selection of k for KNN

To achieve high accuracy, selecting the best number of neighbors(k) is necessary. I iterated k in the range between 1 and 25. Then plot the figure 8 to observe the highest accuracy in relation to k. It shows that 6 is the best number of neighbors.

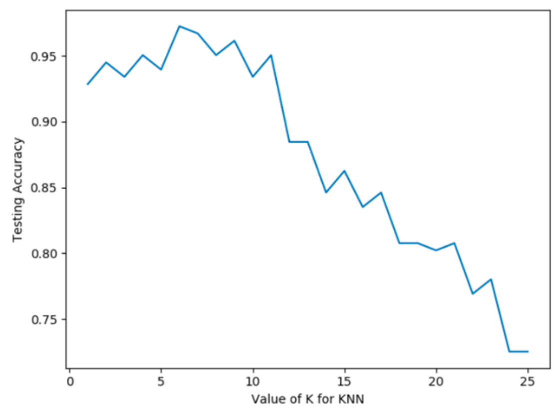


Fig. 8. value k for KNN verus accuracy

IV. RESULTS AND DISCUSSION

To evaluate the performance of the implementation with 20% training set and 80% as test set, classification report is very intuitionistic.

	precision	recall	f1-score	support
0.0	0.96	1.00	0.98	132
1.0	1.00	0.90	0.95	50
accuracy			0.97	182
macro avg	0.98	0.95	0.96	182
weighted avg	0.97	0.97	0.97	182

Fig. 9. classification report for KNN with k = 6

	precision	recall	f1-score	support
0.0	0.90	0.97	0.93	132
1.0	0.90	0.70	0.79	50
accuracy			0.90	182
macro avg	0.90	0.83	0.86	182
weighted avg	0.90	0.90	0.89	182

Fig. 10. classification report for SVM

From Figure 9 and Figure 10, it is obvious that KNN with the best number of neighbors performs better than SVM. KNN's accuracy is as high as 0.97 and class Arabidopsis's recall is 1.00, which means all Arabidopsis test cases have been classified correctly. I think the reason that class Tobacco performs not as well as Arabidopsis is due to that the size of Tobacco is less than half of Arabidopsis.

Because the dataset used for the individual part is not big, the efficiency comparison between KNN and SVM does not show up. If it is a very large dataset, I think SVM will perform better than KNN in general.

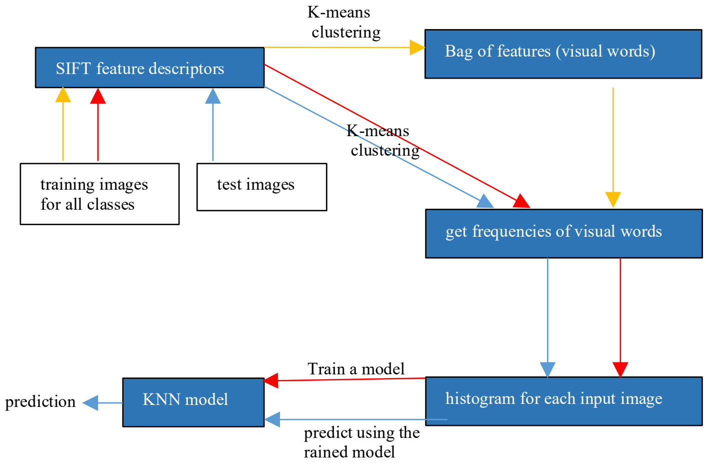


Fig. 11. general steps [3]

In conclusion, Figure 11 is the plot for general steps of my implementation. I use Bag-of-Words for image representation, SIFT for feature extraction and KNN for pattern recognition after comparison. It performs well generally.

REFERENCES

[1] Wang, Shuai and Zhendong Su. "Metamorphic Testing for Object Detection Systems." *ArXiv* abs/1912.12162 (2019), pp.1-2.

[2] https://en.wikipedia.org/wiki/Feature_extraction

[3] UNSW COMP9517 Week 3 - Feature Representation - Part 1 (20T3), slide 60-68

[4] A. Muthukumar, A. Kavipriya, A biometric system based on Gabor feature extraction with SVM classifier for Finger-Knuckle-Print, Pattern Recognition Letters,Volume 125, 2019, pp.150-156, ISSN 0167-8655.