Deep Learning
Assignment 1

Ting Chun Yeh
NM6124012
National Cheng Kung University
babytiffany2000@gmail.com

***Abstract*—This assignment is all about image classification. We are required to build a system using at least three different classification models and three different image feature extraction methods. Furthermore, after finishing experiments, we need to compare the performance of different combinations, and discuss about the effects of the feature extraction methods and the classification models to the performance.**

Keywords—feature extraction, image classification

I. INTRODUCTION

Unlike features that are already written down in words, we need to do some transformations to the photo in order to extract the features of the images. Unlike neural network, if we want to use traditional machine learning classification models for image classification, we need to do a step first prior to training and testing. And which is we need to do image feature extraction first before training and testing. There are tons of image feature extraction methods and tons of different classification models. However, different combination would have different effect to the results. In this assignment, I will implement four different classification models, which are K-Nearest Neighbors, Support Vector Machine, CatBoost, and Decision Tree. As for image feature extraction methods, I will be using SIFT, ORB, and BRISK. The methods and the performance results will be discussed in the later section.

II. METHOD

A. SIFT

SIFT, stand for Scale-invariant feature transform. It is an algorithm to detect and describe local features of the images invariant to scale. It consists several stages, including scale-space extrema detection, keypoint localization, orientation assignment, and descriptor generation. Localized feature description and detection help recognize objects, and this is why it is helpful for classification. For it can catch the similarity between features of images despite the size of the images and whether the object is rotate or not.

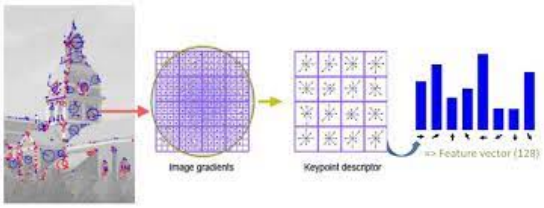

Figure 1. SIFT Descriptor Illustration

B. ORB

ORB, stand for Oriented Fast and Rotated BRIEF. ORB builds on the FAST keypoint detector and the BRIEF descriptor, and this is ORB's key innovation. FAST efficiently detects keypoints by identifying corners or edges in an image. BRIEF then generates binary descriptors for these keypoints, representing local image patches using a set of pre-defined binary test. Due to FAST features not having an orientation component and multiscale features, ORB uses a multi scale image pyramid which is a multiscale representation of a single image that consist of sequences of images all of which are versions of the image at different resolutions. This way ORB is partial scale invariant.

Figure 2. Oriented Fast and Rotated BRIEF.

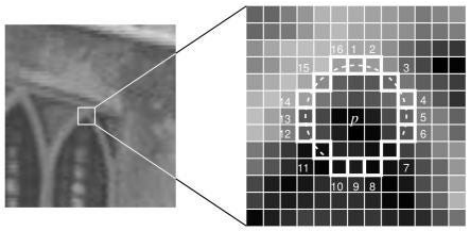

Figure 3. FAST Algorithm for corner detection.

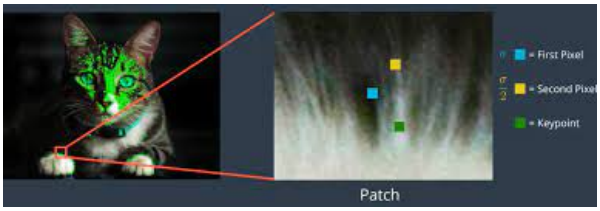

Figure 4. BRIEF selects a random pair and assign the value to them.

C. BRISK

BRISK, stand for Binary Robust Invariant Scalable Keypoints. It is scale invariant and rotation invariant. Similar to BRIEF, for capturing the local image information around keypoints. However, unlike BRIEF, which uses random tests for descriptor generation. First of all, BRISK detects keypoints using a scale-space approach, which is similar to SIFT. Once keypoints are detected, BRISK computes a binary descriptor for each keypoint based on the local image gradient information sampled according to its predefined pattern.

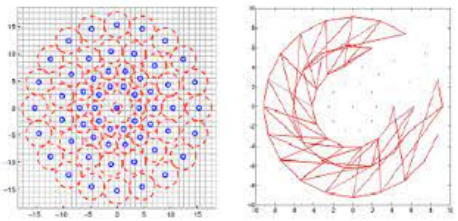

Figure 5. BRISK Descriptor Extraction

D. *K-Nearest Neighbors*

K-Nearest Neighbors, also known as KNN, is one of the well-known classification method. The choice of k is crucial when using the classifier. Which k we choose may affect the results by a lot. KNN is a non-parametric algorithm. It is simple to use, but it can handle quite difficult decision boundaries and is particularly useful when the decision boundary is irregular or hard to define using parametric models. Here, I simply set n_neighbors equals to 5 (default value).

E. *Support Vector Machine*

Support Vector Machine, also known as SVM, is a very popular model even if now neural network almost takes all over the world. The strategy of SVM is to find the optimal hyperplane which is able to separate different classes in the input space while maximizing the margin between the classes. SVM can handle both linearly and non-linearly separable data by using different kernel functions, and which make it effective in high-dimensional spaces.

F. *CatBoost*

CatBoost, one of the classification method from the boost family. It is a gradient boosting library. CatBoost actually stands for Category Boosting, it focus more on categorical variables. CatBoost doesn't require explicit pre-processing steps for categorical variable, instead it can directly deal with categorical features. CatBoost employs the gradient boosting framework. It optimizes a differentiable loss function by iteratively adding weak learners (decision trees) to the ensemble. Additionally, CatBoost implements several advanced techniques which improves the learning process by consider the ordering of categorical features during tree construction.

G. *Decision Tree*

Decision Tree, a model that builds tree-like structure based on the features by recursively splitting the data based on feature attributes. The tree structure consists of nodes, and branches. Here, the nodes are the features. At each node, the model will make the rule of how the data is going to be split into subsets. The rules are typically made based on criteria such as Gini Impurity or Information Gain (if it's a classification task). The data will be split recursively until there is no need to split the data, meaning all the samples have been classified and get a predicted label.

III. EXPERIMENT

For the experiment, I use all the train data as training dataset, and all the test data as testing dataset. For the model system, I use three different image feature extraction methods, SIFT, ORB, BRISK, along with four different classification models, K-Nearest Neighbors, Support Vector Machine, CatBoost, Decision Tree. After applying feature extractor, before training, I use K-Means in order to compress the extracted features (here I set k equals to 40). According to the instruction, I will be using accuracy score and f1-score as the metrics to evaluate the performance.

	SIFT	ORB	BRISK
K-Nearest Neighbors	0.5%	0.5%	1%
Suppor Vector Machine	**2%**	0.5%	0.5%
CatBoost	0.5%	0.5%	0.5%
Decision Tree	1%	0.5%	0%

Table 1. Accuracy score of different combination of the image feature extractor and classifier.

	SIFT	ORB	BRISK
K-Nearest Neighbors	0.1667%	0.0058%	0.3429%
Suppor Vector Machine	**1.0929%**	0.0064%	0.125%
CatBoost	0.1667%	0.0064%	0.3333%
Decision Tree	0.5333%	0.0064%	0%

Table 2. F1-score of different combination of the image feature extractor and classifier.

According to the experiment results, we can see even though the results are not pretty good in general; however, there is still a bit disparity between the results. For this dataset, the model combination which using SIFT as the feature extractor and Support Vector Machine as the classifier had the better performance in both accuracy and f1-score.

IV. CONCLUSION

When doing image classification using machine learning classification models, feature extraction is necessary. And which feature extractor we choose will affect the information the training model could get. The performance of different combination of a model would differ according to the dataset, the situation, the environment, and a lot of different factors. Here I would say for this TinyImageNet dataset under the situation that using all the training set I get as training data, the model which uses SIFT as feature extractor and Support Vector Machine as classifier will have the best performance of all. Nonetheless, I can't deny that the performance is still not pretty good whether for this model or overall.

REFERENCES

1. https://zx7978123.medium.com/圖像相似度算法-移動偵測-入侵偵測-ae161d355537#:~:text=SIFT是一種機器視覺,、尺度、旋轉不變數%E3%80%82&text=局部性特徵的描述,的大小和旋轉無關%E3%80%82
2. https://velog.io/@dusruddl2/Photogrammetry-8-1.-Visual-Features-Desciptors-SIFT-BRIEF-ORB
3. https://medium.com/@deepanshut041/introduction-to-orb-oriented-fast-and-rotated-brief-4220e8ec40cf
4. https://medium.com/analytics-vidhya/feature-matching-using-brisk-277c47539e8
5. https://www.researchgate.net/figure/BRISK-descriptor-extraction_fig1_287196454
6. ChatGPT

Identify applicable funding agency here. If none, delete this text box.

V. Ablation Study

After using SIFT, ORB, and BRISK as feature extractor, here I use ResNet50 as feature extractor, then combine with classification models, K-Nearest Neighbors, Support Vector Machine, CatBoost, and Decision Tree, just like above, in order to see whether there will be any differences between these two kind of methods.

ResNet50 is a different kind of feature extraction method from all the above (SIFT, ORB, and BRISK). As we all know, ResNet50 is a type of convolutional neural network. It consists of 50 layers of neural network weights, including convolutional layers, pooling layers, and fully connected layers. Here I chose to let the model be pre-trained on ImageNet, as a result, the network has already extracted lots of rich and meaningful features for the images of the dataset. To use ResNet50 as a feature extractor, we typically remove the fully connected layers at the end of the network. We input these images to the ResNet50 without fully connected layer then the output will be the extracted features. After that, we input these extracted features into other classification models like KNN or SVM, then we can get the results.

	Accuracy	F1-score
K-Nearest Neighbors	36%	29.1595%
Suppor Vector Machine		
CatBoost		
Decision Tree		

Table 3. Performance under ResNet50 as feature extractor

Here we can see that the performance is actually much better comparing to the results under SIFT, ORB, or BRISK as feature extractor.

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an MSW document, this method is somewhat more stable than directly inserting a picture.

To have non-visible rules on your frame, use the MSWord “Format” pull-down menu, select Text Box > Colors and Lines to choose No Fill and No Line.