Image Classification with Multiple Models and Features

*Deep Learning HW1

Liang-Jen Huang
Institute Of Data Science, NCKU
March 16, 2024

***Abstract*—The goal of this assignment is to explore different image feature extraction methods and observe their performance across various models. Here, I utilized four feature extraction methods: FAST+BRIEF, ORB, Color Histogram, and HOG, along with four machine learning models: SVM, KNN, RF, and AdaBoost, for the task of image classification. The code is provided at: https://github.com/edogawa-liang/DL-2024spring/tree/main/HW1**

I. Introduction

With the rapid development in the fields of machine learning and computer vision, image classification has become an important research direction. It has wide applications in numerous areas such as facial recognition, medical image analysis, and autonomous driving. The key to image classification lies in how to extract effective features from the original images and use them in classification models. Good features can not only improve the accuracy of classification but also enhance the generalization ability of the model.

II. The Proposed Method

Since this assignment uses common machine learning models, the focus here is on introducing the feature extraction methods. Among the four methods, except for the Color Histogram, the rest do not support color images and require conversion to grayscale first.

FAST + BRIEF. Here, I used FAST to quickly detect corners as feature points and BRIEF to describe these feature points.

FAST (Features from Accelerated Segment Test) is a corner detection algorithm that can quickly identify corner points, i.e., points in the image where local feature changes are significant. The principle is to select a pixel in the image, set a threshold, and check whether the pixels within its circular neighborhood are significantly brighter or darker than the center. If the degree of brightness or darkness exceeds the threshold, the point is considered a corner. The main advantage of FAST is its speed, but it is sensitive to noise and the detected corners may change when the image is scaled.

BRIEF (Binary Robust Independent Elementary Features) is a fast feature descriptor algorithm. For each corner point detected by the FAST algorithm, the BRIEF algorithm analyzes the area around the point. It samples pixel pairs in a random or predetermined pattern and generates a binary string representing the comparison result of each pair as the feature descriptor. For example, if the first pixel is brighter than the second, it is recorded as 1, otherwise as 0. Eventually, each corner point will receive a binary descriptor string.

ORB. ORB (Oriented FAST and Rotated BRIEF) is an algorithm that combines FAST keypoint detection and BRIEF feature descriptors. It improves accuracy and robustness of feature matching by calculating the orientation of feature points through FAST for rotational invariance and using image pyramids for some scale invariance. When computing feature descriptors with BRIEF, the sampling pattern is also rotated according to the orientation of each keypoint. These improvements enhance the accuracy and robustness of feature matching.

Color Histogram. Since each pixel in an image is composed of values from the R, G, and B channels, this method draws a histogram of the distribution of three colors in the photo to view the distribution of colors as the image feature. The advantage of this method is that the image feature does not change with rotation or scaling, and the calculation process is simple and fast. However, this calculation method only considers the proportion of colors, not the position and shape of features, and changes in the image's brightness can lead to different Color Histograms.

HOG. The Histogram of Oriented Gradients (HOG) captures the shape of local objects in an image by describing the distribution of gradient orientations. Specifically, the image is first divided into multiple overlapping small areas called Cells. Then, several adjacent Cells are grouped to form a Block. Within each Cell, the direction of pixel gradients is calculated, and these directional informations are plotted into histograms, which are combined to form the Block. Since direct calculation of gradient orientations can be affected by changes in brightness and shadows, each Block undergoes contrast normalization to enhance the model's robustness to variations in lighting.

III. Experiments

The data for this experiment includes 99600 images in the training set and 200 images in the test set, across 200 categories. Initially, the images were resized from 64*64 to 256*256 dimensions before using the above-mentioned feature extraction methods. The related hyperparameters are shown in

TABLE I
SETTINGS OF FEATURE EXTRACTOR

Feat. Extractor	**Setting**	**Dim.**
FAST + BRIEF + pooling	npoints: 256 (default) Kmeans: k=32	32
ORB + pooling	nfeatures: 500 (default) Kmeans: k=32	32
ColorHist	Bins of R * G * B = 8	512
HOG	WinSize: (128, 128) BlockSize: (64, 64) BlockStride: (32, 32) CellSize: (64, 64) Nbins: 5	405

Table I. Then, the SVM, KNN, RF, and AdaBoost models were used for the image classification task, without special hyperparameter selection; default model parameters were used, and Accuracy and F1 score were used as evaluation metrics.

In this experiment, due to the large computational load of SVM, regardless of the feature extraction method used, the SVM model was preceded by PCA dimensionality reduction to 32 dimensions before being input into the model. The experiment results are shown in Table II. Overall, the model performance was not very good, with models using ColorHist feature extraction generally outperforming those using FAST+BRIEF, ORB, and HOG in terms of Accuracy and F1 score. Random Forest combined with ColorHist achieved the highest Accuracy (0.105) and F1 score (0.0722). These results suggest that, for this task, color-based feature extraction methods may provide more information than other local binary (FAST + BRIEF, ORB) or gradient-based descriptors (HOG), and Random Forest and PCA+SVM generally performed better. However, apart from the test set performance, the training set accuracy (Training Acc) shows that most models also struggled to effectively learn features during the training stage, with Random Forest experiencing severe overfitting. This not only illustrates the difficulty traditional machine learning models face in handling image classification tasks but also implies that the feature extraction step may not have fully captured information helpful for classification.

Table III presents the time spent on feature extraction and model training phases of this experiment. Among the feature extraction methods, Color Histogram was the fastest, consistent with the discussion in Section 2 that color distribution detection as a feature has lower computational complexity. Due to the need for further Kmeans clustering with methods processing local features (FAST+BRIEF and ORB), more time was spent. Regarding model training, even with PCA dimensionality reduction, SVM required far more time than other models, reflecting its higher computational demand.

TABLE II
RESULTS

Feat. Extractor	**Model**	**TrAcc**	**TestAcc**	**TestF1**
FAST + BRIEF + pooling	PCA+SVM	0.136	0.020	0.010
	KNN	0.238	0.005	0.005
	RF	0.999	0.005	0.005
	AdaBoost	0.030	0.005	0.000
ORB + pooling	PCA+SVM	0.103	0.010	0.004
	KNN	0.222	0.015	0.010
	RF	0.997	0.010	0.004
	AdaBoost	0.024	0.005	0.003
ColorHist	PCA+SVM	0.126	0.080	0.052
	KNN	0.268	0.030	0.019
	RF	1.000	0.105	0.072
	AdaBoost	0.045	0.015	0.008
HOG	PCA+SVM	0.187	0.060	0.044
	KNN	0.213	0.015	0.007
	RF	0.963	0.045	0.026
	AdaBoost	0.026	0.020	0.011

TABLE III
THE TIME REQUIRED FOR FEATURE EXTRACTION AND TRAINING

Feat. Extractor	**Model**	**Dim.**	**Feat.(s)**	**Tr.(s)**
FAST + BRIEF + pooling	PCA+SVM	$32 \rightarrow 32$	560	8245
	KNN	32		215
	RF	32		186
	AdaBoost	32		86
ORB + pooling	PCA+SVM	$32 \rightarrow 32$	613	10520
	KNN	32		452
	RF	32		245
	AdaBoost	32		87
ColorHist	PCA+SVM	$512 \rightarrow 32$	21	8990
	KNN	512		26
	RF	512		396
	AdaBoost	512		187
HOG	PCA+SVM	$405 \rightarrow 32$	210	8454
	KNN	405		14
	RF	405		1333
	AdaBoost	405		504

IV. CONCLUSIONS

In this assignment, various methods were used to extract features from images, and common machine learning models were employed to learn these features and perform image classification. The experimental results were unsatisfactory, highlighting the limitations of feature extraction methods and traditional machine learning. Future attempts could explore deep learning methods, such as convolutional neural networks (CNN), which can automatically extract features through filters and integrate feature extraction with classification tasks in an end-to-end training framework, potentially improving image classification performance.

REFERENCES

[1] OpenCV Python Tutorials, Available online: https://opencv-python-tutorials.readthedocs.io/zh/latest/

[2] OpenCV-with-Python, Available online: https://github.com/chewbacca89/OpenCV-with-Python/blob/master/Lecture%205.5%20-%20SIFT%2C%20SURF%2C%20FAST%2C%20BRIEF%20%26%20ORB.ipynb

[3] Bag-of-Words Model in NLP, Available online: https://medium.com/@derekliao_62575/nlp%E7%9A%84%E5%9F%BA%E6%9C%AC%E5%9F%B7%E8%A1%8C%E6%AD%A5%E9%A9%9F-ii-bag-of-words-%E8%A9%9E%E8%A2%8B%E8%AA%9E%E8%A8%80%E6%A8%A1%E5%9E%8B-3b670a0c7009

[4] Udacity Computer Vision, Available online: https://medium.com/chiukevin0321/%E4%BA%BA%E8%87%89%E8%BE%A8%E8%AD%98-face-detection-face-recognition-7ba98aaf1a02