Image Classification with Multiple Models and Features

Jen-Lung Hsu
Institute of Data Science, National Cheng Kung University
Tainan, Taiwan
RE6121011@gs.ncku.edu.tw

Abstract—**This study investigates the performance of various classification models and feature extraction methods for image classification tasks. We explore K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest as models, and Scale-Invariant Feature Transform (SIFT), Oriented FAST and Rotated BRIEF (ORB), and Binary Robust Invariant Scalable Keypoints (BRISK) as feature extraction techniques. ResNet18 is utilized specifically for feature extraction purposes, generating feature maps to characterize images. Our experiments provide a thorough analysis of different model-feature combinations, offering insights into their performance across multiple metrics. These findings contribute to a deeper understanding of image classification methodologies and provide valuable guidance for future research and development.**

Keywords—classification models, feature extraction methods

I. Introduction

In this report, we present the methodology, experiments, and results of an image classification system utilizing various classification models and feature extraction techniques. Our objective is to analyze the impact of different models and features on the performance of image classification tasks.

II. Methodology

A. Selection of Classification Models

We implemented three classification models: K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest.

1) KNN is a simple yet effective algorithm that classifies objects based on the majority class among their k nearest neighbors in the feature space. Its time complexity is $O(n \log(n))$, where n is the number of training samples.

2) SVM is a powerful model for classification tasks, particularly suited for high-dimensional spaces, by finding the optimal hyperplane that maximizes the margin between classes. Its time complexity ranges from $O(n^2)$ to $O(n^3)$, depending on the number of training samples.

3) Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes as the prediction. Its time complexity for training is $O(mn \log(n))$, where m is the number of trees and n is the number of training samples.

B. Feature Extraction Techniques

1) Scale-Invariant Feature Transform (SIFT): SIFT detects keypoints in an image and generates descriptors that are invariant to scale, rotation, and illumination changes. It achieves this by using a scale-space representation and applying gradient-based edge detection techniques. The keypoints identified by SIFT are robust to transformations, making them suitable for tasks like object recognition and image matching across different perspectives.

2) Oriented FAST and Rotated BRIEF (ORB): ORB is designed to be a computationally efficient alternative to SIFT and SURF. It combines the FAST keypoint detection algorithm with the BRIEF descriptor, generating binary descriptors for key points in images. By using binary descriptors, ORB reduces memory usage and speeds up feature matching processes, making it well-suited for real-time applications and scenarios with large datasets.

3) Binary Robust Invariant Scalable Keypoints (BRISK): BRISK is a binary descriptor that builds upon the concepts of the FAST keypoint detector and uses a pyramid-based scale space to capture features across different scales. Unlike traditional gradient-based descriptors, BRISK employs a gradient orientation estimation method that enhances robustness to varying lighting conditions and viewpoint changes. BRISK's combination of speed, robustness, and scalability makes it suitable for applications such as image stitching, panorama creation.

4) Learning-based feature extraction using RESNET18: RESNET18 is a deep convolutional neural network (CNN) architecture that has been pre-trained on large-scale datasets like ImageNet. By leveraging the learned representations from RESNET18, we can extract high-level features from images that capture complex patterns and semantic information. The deep features extracted by RESNET18 are transferable and can be fine-tuned or used directly for various computer vision tasks such as image classification, object detection, and image retrieval.

C. Data Preprocessing

Prior to feature extraction, we performed data preprocessing steps including contrast enhancement, sharpening, and noise reduction. These steps aim to improve the quality of input images and enhance the discriminative features for classification.

Due to the variable feature lengths extracted from each image, we standardized the feature shape to a fixed length by discarding excess features and padding with zeros for shorter ones.

D. Justification of Feature Selection

We selected SIFT, ORB, and BRISK as they are robust to various transformations and lighting conditions, making them suitable for diverse image datasets. Additionally, we incorporated RESNET18-based features to leverage the power of deep learning representations, which can capture hierarchical features from raw image data.

III. EXPERIMENTS

We conducted experiments using the provided training and testing datasets with and without preprocessing steps. Each model was trained and tested with different combi-nations of feature sets, including SIFT, ORB, BRISK, and RESNET18-based features. We evaluated the performance using F1-Score and Accuracy metrics.

IV. RESULTS AND DISCUSSION

Below are the experimental results for the 12 combinations of three models and four feature extraction methods. The computational speed for each combination is also provided. Additionally, we conducted another set of 12 experiments without image preprocessing. Additionally, we conducted experiments using a dataset containing 10 categories instead of the entire dataset. This modification was made to reduce computational time while still maintaining a diverse set of classes for robust evaluation. Therefore, the experimental results presented below are based on the reduced dataset of 10 categories.

TABLE I. THE PERFORMANCE OF KNN USING FOUR DIFFERENT FEATURE EXTRACTION METHODS WITH PREPROCESSING

KNN with preprocessing					
	Accuracy	F1-score	Training Time (Unit: seconds)	Testing Time (Unit: seconds)	Total Time (Unit: seconds)
SIFT	**0.2**	**0.1**	0.0130	0.3159	0.3289
ORB	0.0	0.0	0.0040	0.0658	0.0698
BRISK	0.1	0.04	0.0199	0.1326	0.1525
RESNET	0.1	0.067	**0.0020**	**0.0269**	**0.0289**

TABLE II. THE PERFORMANCE OF SVM USING FOUR DIFFERENT FEATURE EXTRACTION METHODS WITH PREPROCESSING

SVM with preprocessing					
	Accuracy	F1-score	Training Time (Unit: seconds)	Testing Time (Unit: seconds)	Total Time (Unit: seconds)
SIFT	**0.2**	**0.2**	345.4830	0.4664	345.9494
ORB	0.1	0.1	11.0927	0.0309	11.1236
BRISK	**0.2**	0.167	289.9884	0.2748	290.2631
RESNET	0.0	0.0	**2.2507**	**0.0060**	**2.2567**

TABLE III. THE PERFORMANCE OF RANDOM FOREST USING FOUR DIFFERENT FEATURE EXTRACTION METHODS WITH PREPROCESSING

Random Forest with preprocessing					
	Accuracy	F1-score	Training Time (Unit: seconds)	Testing Time (Unit: seconds)	Total Time (Unit: seconds)
SIFT	**0.1**	**0.1**	34.8878	0.0070	34.8948
ORB	0.0	0.0	19.7150	0.0080	19.7230
BRISK	**0.1**	0.067	19.2874	0.0080	19.2954
RESNET	**0.1**	0.05	**14.7736**	**0.0080**	**14.7815**

TABLE IV. THE PERFORMANCE OF KNN USING FOUR DIFFERENT FEATURE EXTRACTION METHODS WITHOUT PREPROCESSING

KNN without preprocessing					
	Accuracy	F1-score	Training Time (Unit: seconds)	Testing Time (Unit: seconds)	Total Time (Unit: seconds)
SIFT	**0.2**	**0.1**	0.0171	0.2847	0.3017
ORB	0.1	0.05	0.0040	0.0698	0.0738

BRISK	0.1	0.05	0.070	0.1593	0.1663
RESNET	0.1	0.067	**0.0010**	**0.0379**	**0.0389**

TABLE V. THE PERFORMANCE OF SVM USING FOUR DIFFERENT FEATURE EXTRACTION METHODS WITHOUT PREPROCESSING

SVM without preprocessing					
	Accuracy	F1-score	Training Time (Unit: seconds)	Testing Time (Unit: seconds)	Total Time (Unit: seconds)
SIFT	0.0	0.0	322.1163	0.5105	322.6269
ORB	**0.2**	**0.2**	10.8229	0.0289	10.8518
BRISK	0.1	0.1	228.4167	0.2747	228.6914
RESNET	0.1	0.05	**2.3462**	**0.0060**	**2.3521**

TABLE VI. THE PERFORMANCE OF RANDOM FOREST USING FOUR DIFFERENT FEATURE EXTRACTION METHODS WITH PREPROCESSING

Random Forest without preprocessing					
	Accuracy	F1-score	Training Time (Unit: seconds)	Testing Time (Unit: seconds)	Total Time (Unit: seconds)
SIFT	0.1	0.1	25.7144	0.0050	25.7194
ORB	0.0	0.0	13.2695	0.0030	13.2725
BRISK	**0.2**	**0.2**	13.6219	0.0030	13.6249
RESNET	0.0	0.0	**10.8010**	**0.0030**	**10.8040**

From the table above, we can observe the following patterns. With preprocessing, the SIFT feature extraction method performs well across all three models, but it also has the longest processing time. Without preprocessing, the performance varies: SIFT is better for KNN, ORB is better for SVM, and BRISK is better for Random Forest. Interestingly, regardless of preprocessing, the RESNET feature extraction method consistently has the fastest processing time. This might be attributed to its nature as a learning-based feature extraction method, providing discriminative features that can be quickly classified by the models.

However, overall, the performance of all combinations seems subpar. This could be due to flattening the extracted two-dimensional features into one-dimensional vectors, potentially losing spatial information crucial for classification. It could also be attributed to the fact that I discarded excess features and padded shorter features with zeros to address the issue of variable feature lengths across images. Exploring alternative methods that preserve more information and account for spatial information loss during feature processing may lead to improved performance.

V. CONCLUSION

In summary, our experiments highlighted the effectiveness of the SIFT feature extraction method with preprocessing across various models, albeit with longer processing times. ORB and BRISK showed promise without preprocessing, emphasizing the need for tailored preprocessing strategies. RESNET's efficiency suggests its potential for rapid image classification.

Moving forward, future work involves exploring methods to preserve spatial information, optimizing models through cross-validation, finding more reasonable padding ways, and leveraging ensemble techniques for improved accuracy. Additionally, advanced deep learning architectures and data augmentation can enhance model robustness and generalization.

REFERENCES

[1] Lowe, D. G. "Distinctive Image Features from Scale-Invariant Keypoints". International Journal of Computer Vision, 2004.

[2] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. "ORB: An efficient alternative to SIFT or SURF". In IEEE International Conference on Computer Vision (ICCV), 2011.

[3] Leutenegger, S., Chli, M., & Siegwart, R. Y. "BRISK: Binary Robust Invariant Scalable Keypoints". In IEEE International Conference on Computer Vision (ICCV), 2011.

[4] ChatGPT