DEEP LEARNING ASSIGNMENT 1

Pei-Chun Chen

Department of Statistics, National Cheng Kung University
R26121073
lres50804@gmail.com

ABSTRACT

This assignment focuses on the simple processing and comparison of image data, necessitating the integration of at least three different feature extraction methods within the realm of image processing. It requires the use of classification models: K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and one additional model of my choice for training and testing purposes. The ultimate goal is to compare the performance of these various models and to discuss the findings. Through this comparative analysis, I aim to elucidate the strengths and weaknesses of each method in handling image data, providing valuable insights into the most effective techniques for image classification.

Keywords- Image Feature Extraction, Image Classification, Machine Learning Algorithms, Feature Descriptor Techniques.

GitHub-`https://github.com/Patrick72Chen/112-2_Deep_learning_Assignment1`

1. INTRODUCTION

In this assignment, I utilized four distinct image feature extraction techniques: Binary Robust Invariant Scalable Keypoints (BRISK), Oriented FAST and Rotated BRIEF (ORB), Scale-Invariant Feature Transform (SIFT), and Histogram of Oriented Gradients (HOG). Subsequently, these features were input into three classification models for training and prediction purposes: K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and eXtreme Gradient Boosting (XGBoost). My analysis focuses on their differences, performance outcomes, and the underlying principles of each feature extraction technique. Moreover, I explore how these principles impact the model's efficacy in accurately classifying and understanding image data. This comprehensive approach allows for a nuanced evaluation of how each technique contributes to the overall performance of the classification models, providing insights into the most effective strategies for image-based machine learning applications.

2. METHODOLOGY

In this section, I will delineate the methodologies of four prominent image feature extraction techniques: BRISK, ORB, SIFT, HOG. Each method will be explored in terms of its foundational concepts and operational mechanisms, providing a foundation for understanding their unique contributions to the field of image processing.

2.1. Feature Extraction - BRISK

Firstly, BRISK, a method designed for efficient feature detection and description, will be examined. Its methodology, which involves using a fast and robust algorithm for keypoint detection, description, and matching, highlights its utility in real-time applications.

The BRISK method's effectiveness is evident in the provided visual (see Figure 1), showcasing a grid with identified keypoints indicated by blue circles with red crosshairs. This demonstrates BRISK's precision in pinpointing relevant features for image analysis. The distribution of keypoints across the image highlights the method's robust feature detection, crucial for reliable image matching and recognition in dynamic settings.[1]

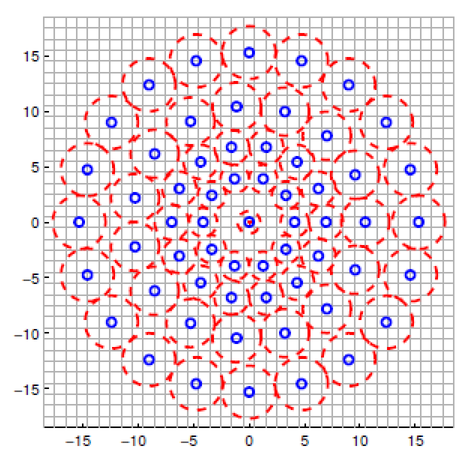

Fig. 1: Keypoint Distribution Pattern by BRISK Algorithm.

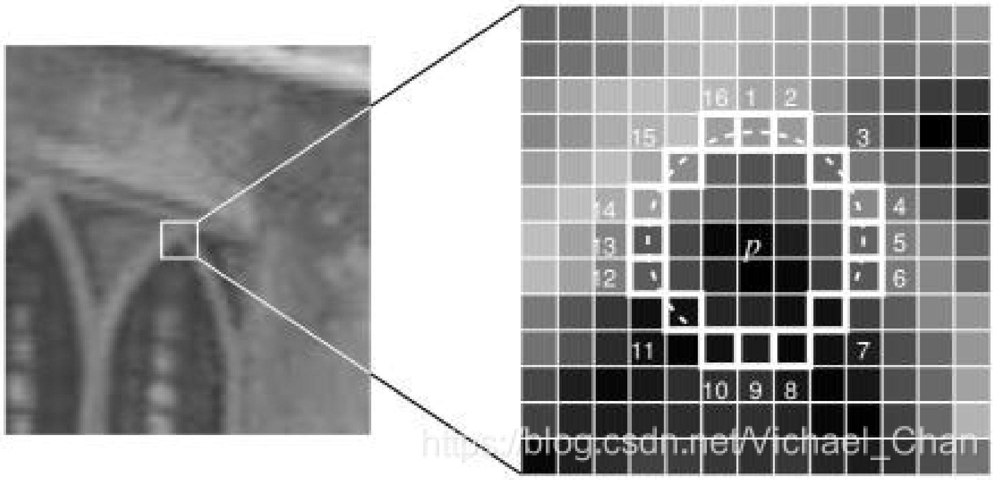

Fig. 2: ORB Feature Detection and Intensity Centroid Localization.

2.2. Feature Extraction - ORB

Secondly, the ORB technique will be analyzed. As a fusion of FAST keypoint detector and BRIEF descriptor with some modifications to enhance performance, ORB stands out for its speed and efficiency, especially in terms of rotation invariance and noise resistance.

The ORB technique, an optimized combination of the FAST keypoint detector and the BRIEF descriptor, offers quick and efficient feature identification and description, excelling in rotation invariance and noise reduction. The accompanying image (see Figure 2) illustrates ORB's feature detection on a detailed portion of an image. It zooms in on a specific area, revealing the algorithm's grid-based approach to pinpoint and rank features based on their intensity levels. Each square, numbered for reference, represents a distinct feature with varying degrees of importance, showcasing ORB's method of creating binary descriptors that are both rotation-invariant and resistant to noise, which are essential qualities for robust real-time image processing.[2]

2.3. Feature Extraction - SIFT

Renowned for its capability to detect and describe local features in images, SIFT's methodology allows for the extraction of features that are invariant to scale, rotation, and illumination changes, making it particularly valuable for object recognition tasks.

The SIFT algorithm commences by crafting a 'scale space' through Gaussian blurring of the original image at various scales, as indicated by the 'Gaussian' labels(see Figure 3). This process ensures that the key locations identified are unaffected by changes in scale and orientation.

Then, to efficiently detect stable keypoint locations in the scale space, the Difference of Gaussian (DoG) function is applied. This is shown in the second series of images labeled 'Difference of Gaussian (DOG)'. The DoG images are obtained by subtracting one blurred image from another image with a slightly different blur level, within the same octave. An octave in this context refers to a set of images within the scale space where the size of the blur doubles.

Extrema in the DoG images, which are potential keypoints, are then located as they will stand out against this background. The keypoints are chosen based on measures of their stability. This step is vital because it allows SIFT features to be resistant to changes in scale, noise, and illumination, which makes them very robust for various tasks in computer vision such as image matching, object recognition, and scene reconstruction.

Lastly, the diagram uses arrows to guide through the progressive scales within each octave, as well as ellipses to suggest the repetition of this process over multiple octaves. This iterative method continues until image resolution is too low for further analysis, emphasizing SIFT's multi-scale approach to feature detection.[3]

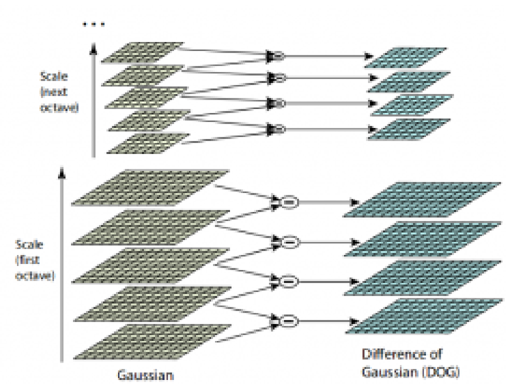

Fig. 3: Scale Space and Keypoint Detection in SIFT: Gaussian Blurring and Difference of Gaussian (DoG).

2.4. Feature Extraction - HOG

This method is centered around counting occurrences of gradient orientation in localized portions of an image. It is particularly effective for the task of object detection in the context of computer vision, owing to its ability to capture edge and texture information while being invariant to geometric and photometric transformations.

The depicted method is Histogram of Oriented Gradients (HOG), which is a feature descriptor used for object detection in computer vision(see Figure 4). This technique analyzes the structure of an image by examining the distribution and direction of gradients, which represent the edges and textures. The process begins with the input image, where a detection window is defined. This window is then subjected to normalization to reduce sensitivity to lighting variations and shadowing.

Next, the image gradients are computed, which capture edge directions and intensities. These gradients are quantified within localized cells, accumulating votes for gradient orientation to form a histogram. The collection of these histograms represents the distribution of directions for the image gradients within the cell.

For better accuracy, contrast normalization is performed over blocks of cells, which helps to further mitigate the effects of illumination and shadowing. The blocks are allowed to overlap, ensuring comprehensive coverage and robustness against spatial shifts.

The final step in the HOG descriptor is compiling these histograms from all the blocks within the detection window into a feature vector. This vector characterizes the local shape and texture around the detection window and can be used to train a machine learning model for tasks such as human detection, by comparing the vector to known HOG patterns associated with the object of interest.[4]

3. EXPERIMENTS AND RESULT

In this experiment, three distinct classification models were employed: K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and eXtreme Gradient Boosting (XGBoost), which were subjected to both training and predictive analysis. This inquiry harnessed four feature extraction techniques, namely BRISK, ORB, SIFT, and HOG.

Table 1: Comparative Accuracy of Three Classification Models Utilizing Four Feature Extraction Methods

	XGBoost	K-NN	Support Vector Machine
BRISK	0.2	0.6	0.2
ORB	0.6	0.6	0.6
SIFT	0.4	0.4	0.8
HOG	0.0	0.6	0.2

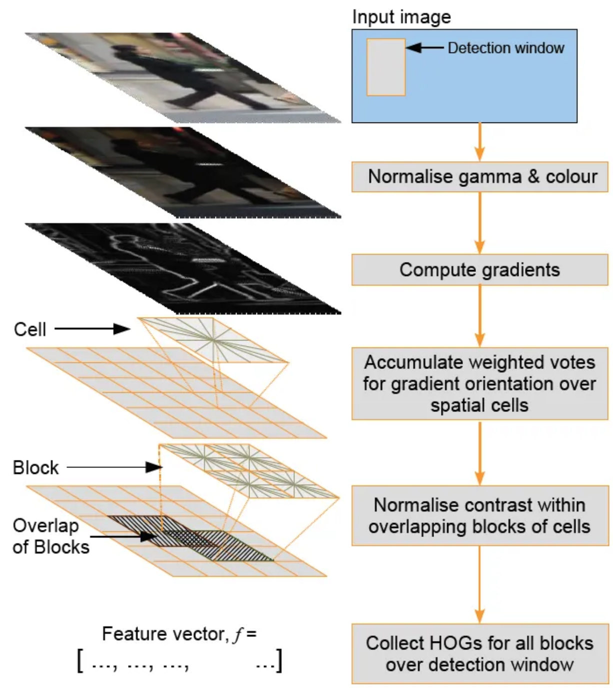

Fig. 4: Flowchart of the HOG Feature Calculation Process

Table 2: Comparative F1-Score of Three Classification Models Utilizing Four Feature Extraction Methods

	XGBoost	K-NN	Support Vector Machine
BRISK	0.1334	0.5000	0.1334
ORB	0.5334	0.5334	0.5334
SIFT	0.2667	0.3000	0.7334
HOG	0.0000	0.4667	0.0667

The efficacy of these models was gauged using two principal metrics: F1-Score (table 2) and accuracy (table 1). To streamline the experimental process, only a subset comprising the first five categories was selected as the sample space. Furthermore, for the k-means clustering algorithm, a configuration of k=50 clusters was stipulated to facilitate a more focused and efficient clustering outcome.

From table 1 and table 2, in this analysis, a discernible under-performance is noted when Histogram of Oriented Gradients (HOG) features are applied in conjunction with the eXtreme Gradient Boosting (XGBoost) model. The subpar efficacy can potentially be ascribed to a dissonance between the gradient-focused features of HOG and the more abstract feature assimilation processes inherent to XGBoost.

Furthermore, BRISK and HOG display similar performance trends, with K-NN showing more promising results as opposed to XGBoost and SVM. This could be due to the nature of the K-NN algorithm, which is non-parametric and operates based on locality in feature space. Given that both BRISK and HOG encapsulate local gradient information, K-NN may be better positioned to recognize and classify these patterns effectively, particularly when dealing with a limited number of classes or when the decision boundaries between classes are not linear.

Finally, it is noted that the combination of SIFT features with the SVM model achieves the best performance. This superior result is likely owing to SIFT's robustness in feature extraction, which includes scale and rotation invariance, paired with SVM's effectiveness in finding the optimal hyperplane that distinctly classifies the features. SIFT provides a rich description of keypoint localities, which SVM may utilize more effectively through its margin maximization approach, especially in higher-dimensional spaces where SVM excels.

4. REFERENCES

[1] *https://juliaimages.org/ImageFeatures.jl/stable/tutorials/brisk/#:~:text=The%20BRISK%20(Binary%20Robust%20Invariant,is%20smoothed%20using%20gaussian%20smoothing*.

[2] *https://blog.csdn.net/Vichael_Chan/article/details/101427437*.

[3] *https://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/*.

[4] *https://medium.com/lifes-a-struggle/hog-svm-c2fb01304c0*.