جامعة بيرزيت

BIRZEIT UNIVERSITY

Electrical and Computer Engineering Department
Computer Vision ‑ ENCS5343
Assignment #2

Sondos Shahin 1200166
Instructor: Dr. Aziz Qaroush
Section 2
Date: Dec 22, 2024

Introduction

The aim of this assignment was to explore and apply local feature extraction techniques to identify handwritten texts. Feature extraction algorithms such as SIFT and ORB were evaluated based on accuracy, efficiency, and robustness to variations in the handwritten samples.

The objective of this assignment was to classify the images so that when the model takes an image as an input, it can determine which user wrote the word in the image.

Background

Scale Invariant Feature Transform (SIFT) algorithm is a feature extraction algorithm that outputs keypoints and computes their descriptors. It came to solve the problems of edges and corners detectors. The algorithm has multiple steps for the extraction of keypoints. The first is to construct a scale space. This step aims to make SIFT scale-invariant. The next step is Laplacian of Gaussian approximation. Then finding Keypoints, Eliminating edges and low contrast regions. Then, assigning an orientation to the keypoints, an orientation is assigned to each keypoint to achieve invariance to image rotation. And finally, generating the SIFT features. The problem with this algorithm is that it is computationally expensive [1].

Oriented FAST and Rotated BRIEF (ORB) is an efficient alternative to SIFT in terms of computational cost. ORB is basically a fusion of FAST keypoint detector and BRIEF descriptor with many modifications to enhance the performance. First it uses FAST to find keypoints, then applies Harris corner measure to find top N points among them. It also uses pyramids to produce multiscale-features [2].

"Bag of Words" is a way to simplify object representation as a collection of their subparts. In Computer Vision, we can consider an image to be a collection of image features. By incorporating frequency counts of these features, we can apply the "Bag of Words" model towards images and use this for prediction tasks such as image classification.

Methodology

The first step was downloading, reading and storing the images from the dataset. The AHAWP dataset (Arabic Handwritten Automatic Word Processing) was used for this assignment. The dataset includes 10 unique Arabic words, handwritten by 82 individuals, with each writer contributing approximately 10 samples per word. This resulted in a total of 8,144 word images in the dataset. The number of images in the dataset is not large enough, so the model might overfit, and the error on the testing data would be high. As a result, data augmentation was needed as a step before starting to build the model. Another use of data augmentation is to add generalization to the model, and therefore the model would be more robust to changes in the images. Rotation and scaling were performed on each image in the dataset. All images were then stored in a list as a tuple (image, label), where the label is the user id.

Images in the dataset were shuffled randomly before splitting, to ensure that each image and each user exist in both the training and the testing sets. The next step was to extract the keypoints and descriptors from each image in the training set. This step was done once with the SIFT algorithm, and the next time with the ORB algorithm, in order to compare the performance of each algorithm.

Then, clustering of the descriptors was performed. I used k-means clustering for this step, and the number of clusters was subjective. Too few clusters will oversimplify the features, while too many might overfit. Multiple numbers were used before settling to 1000 clusters, this number balanced between the resultant accuracy and computation time. Descriptors were clustered to 1000 clusters, then each cluster was represented by its center. The bag of words (BoW) was constructed from all the clusters' centers.

For each image in the test set, keypoints and descriptors were extracted using SIFT and ORB each time, then their locations in the BoW were determined by finding the minimum distance between the keypoints and each BoW index. Therefore, for an image's descriptors, the nearest distance was used to determine the correct cluster.

The final step was to choose a classifier to train and test its accuracy on the dataset. In order to make a comparison, more than one classifier needed to be trained and tested. The models included the Naive Bayes classifier, Random Forest and SVM with different kernels. For each model, the accuracy of classification on the dataset was calculated.

Naive Bayes was used because it is a probabilistic classifier that works well on texts. Its implementation is simple and efficient. However, it assumes independence between features.

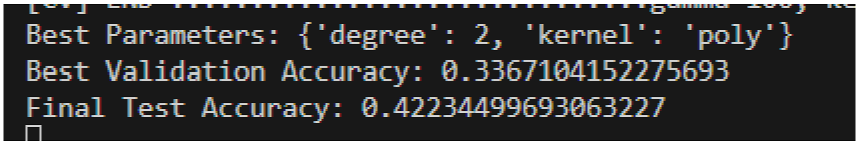
Random Forest is an ensemble learning method that builds multiple decision trees and combines their outputs for classification. It handles large datasets and avoids overfitting. However it requires more computational resources than simpler models. The number of decision trees in the random forest is a hyperparameter. I tried multiple values, but using 100 decision trees resulted in the highest accuracy.

SVM works by finding the hyperplane that best separates the data into different classes in the feature space. Since the dataset is not linearly separable, using the linear kernel will not be effective. Other kernel options include using a polynomial kernel or a radial basis function kernel. Both work well for complex data and high dimensional datasets, but are prone to overfitting and therefore require tuning. In addition to their higher computational time.

Cross validation was done in this step in order to tune the SVM model. A grid search was implemented to find the best model between using a polynomial kernel and an RBF kernel. For the polynomial kernel, the grid search would find the best degree, while for the RBF kernel, grid search finds the best gamma parameter value.

Since our dataset is not large enough, using cross validation would be better than choosing one validation set. Cross validation makes better use of the training data since each sample is used for training and validation. And it provides a more robust estimate of model performance. Moreover, using cross validation reduces the risk of overfitting on the validation set.

The grid search found that for the SVM model, best results would come from using a polynomial kernel with degree 2.



```
Best Parameters: {'degree': 2, 'kernel': 'poly'}
Best Validation Accuracy: 0.3367104152275693
Final Test Accuracy: 0.42234499693063227
```

Results and Analysis

Table 1 and Table 2 show the accuracy of different classifiers, each classifier was used once with SIFT for feature extraction, and another time with ORB. The results show that SIFT is better in terms of accuracy. SIFT produces 128-dimensional, floating-point descriptors, which are robust to changes in scale, rotation, and illumination. ORB produces binary descriptors (32-dimensional, binary vectors), which are much more compact and computationally efficient. However, binary descriptors contain less information and this leads to worse performance than SIFT. ORB's focus is on speed and efficiency rather than descriptor quality. Thus, the lower dimensionality and binary nature of ORB descriptors resulted in worse clustering, and consequently, poorer classification accuracy. SIFT descriptors, due to their higher dimensionality, can better capture the variations and relationships between keypoints in the image. ORB's binary descriptors might cause loss of information during the k-means quantization step, resulting in less discriminative feature representations.

Table 1: Classifiers Accuracy using SIFT

Accuracy of the classifiers using SIFT for feature extraction	
Naive Bayes	28%
Random Forest (100 decision tree)	68%
SVM (Polynomial kernel, degree 2)	42%

Table 2: Classifiers Accuracy using ORB

Accuracy of the classifiers using ORB for feature extraction	
Naive Bayes	21%
Random Forest (100 decision tree)	22%
SVM (Polynomial kernel, degree 2)	23%

The accuracy values did not exceed 70% while using both SIFT and ORB, this could be due to the users’ handwriting, which may be very similar, making it hard to classify correctly. The problem would be better solved using deep learning techniques, which can learn more sophisticated patterns and features from the images.

Figure 1 visualizes the accuracy results from each classifier while using SIFT and ORB for feature extraction.

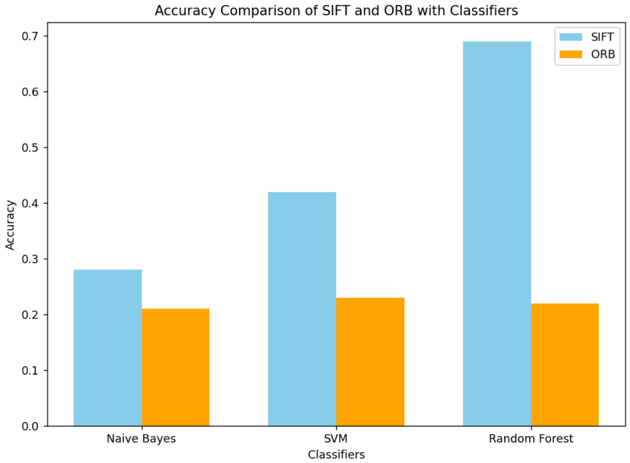


Figure 1- SIFT vs ORB on different classifiers

Table 3 shows that in terms of computational cost, ORB is much better than SIFT. It takes less than half the time that SIFT needs for feature extraction.

Table 3: SIFT and ORB Execution time

Execution Time for Feature Extraction Algorithms	
SIFT	67.5 seconds
ORB	27.2 seconds

SIFT detects a larger number of keypoints, and its descriptors are more stable across images. Whereas ORB detects fewer keypoints because it uses the FAST detector, which can be less sensitive to certain image features. Fewer keypoints means less data for clustering and therefore results in poorer classification performance. Table 4 shows the average number of keypoints that result from each algorithm.

Table 4: Number of keypoints in SIFT and ORB

Average Number of Keypoints per Image	
SIFT	127
ORB	73

Conclusion

In conclusion, the assignment showed that there is a tradeoff between accuracy and computational cost. Using SIFT algorithm for feature extraction gives greater accuracy than using ORB algorithm. On the other hand, ORB is much more efficient than SIFT in terms of computation. Choosing the suitable algorithm for feature extraction is application dependent.

References

[1] https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html

[2] https://docs.opencv.org/4.x/d1/d89/tutorial_py_orb.html