Faculty of Engineering and Technology
Department of Electrical and Computer Engineering
ENCS5343—Computer Vision

Assignment #2—Arabic Handwritten Text Identification Using Local Feature Extraction Techniques

Prepared by:

Ahmad Qaimari —1210190

Yazan AbuAloun —1210145

Instructor: Aziz Qaroush
Date: December 23, 2024

Contents

1 Introduction and Objective

Arabic handwriting is complex due to its intricate curves, varying styles, and connectivity. The Arabic Handwritten Automatic Word Processing (AHWP) dataset offers a valuable resource for studying handwriting-based classification, with applications in forensics, document analysis, and writer identification.This assignment uses local feature extraction techniques, SIFT and ORB, to capture distinct patterns in handwritten Arabic words. These features are then input into a classification pipeline to predict the writer of each word.The goal is to build a system that accurately identifies the writer by exploring the AHWP dataset, extracting features using SIFT and ORB, designing a classification model, and evaluating its performance.

2 Background

Local feature extraction methods, such as SIFT and ORB, are effective at detecting and describing keypoints in images. SIFT is robust to scale and rotation, while ORB is more efficient. Both methods are useful for handwriting analysis. To classify handwriting styles, local features are encoded using techniques like the Visual Bag of Words (BoW) model, enabling machine learning classifiers to work with structured representations.

3 Methodology

3.1 Preprocessing

Preprocessing is a crucial step to standardize the dataset and ensure that the extracted features are consistent and meaningful. In this assignment, two preprocessing techniques were applied:

3.1.1 Resizing

The images were resized to a fixed dimension to ensure consistency across the dataset. This standardization is necessary because feature extraction techniques like SIFT and ORB are sensitive to variations in image size. By resizing all images to the same dimensions, we enable the algorithms to process them effectively, ensuring that the extracted features are comparable across the entire dataset.

3.1.2 Thresholding

A global thresholding technique was applied to binarize the images. This process converts the images into a binary format where pixel values above a certain threshold are set to white, and those below are set to black. This step helps improve the contrast between the handwritten text (foreground) and the background, making the key features of the handwriting more prominent. Thresholding also reduces the impact of background noise, allowing the feature extraction methods to focus on the relevant details of the handwritten words.

To prepare the dataset for analysis, a series of cleaning steps were undertaken. These steps are grouped into the following categories:

3.2 Visual Bag-of-Words Pipeline for Classification

The Visual Bag of Words (VBoW) model, traditionally used in image classification, is adapted here for the task of writer identification in Arabic word text. In this context, the goal is to recognize the specific writer or handwriting style of a given Arabic word. The process integrates feature extraction techniques (such as ORB or SIFT), clustering, and histogramming to convert Arabic words into fixed-size vector representations. These vectors encapsulate the unique visual characteristics of the handwriting, which are then used by machine learning algorithms like SVM to classify the word based on the writer's identity. This approach allows us to effectively capture distinct patterns in the handwriting and identify the writer, even in varied Arabic word forms.

3.2.1 Feature Extraction

At the core of the Visual Bag of Words pipeline are the local feature detectors and descriptors. The `FeatureExtractor` class provides the interface for two different feature extraction methods:

- **ORB (Oriented FAST and Rotated BRIEF)**: This method is a combination of FAST keypoint detection and BRIEF descriptor. It's designed to be fast and efficient, ideal for real-time applications, and works well on images with scale and rotation variations.

- **SIFT (Scale-Invariant Feature Transform)**: This method is designed to detect and describe local features in images. SIFT is robust to changes in scale, rotation, and illumination, making it a powerful tool for image matching tasks.

The `FeatureExtractor` class handles the creation of feature detectors, either ORB or SIFT, based on the user's choice. It uses these detectors to extract keypoints and descriptors from input images. The descriptors are essentially numerical representations of local image regions, capturing unique patterns and structures in the image. These descriptors are later used in the clustering step.

3.2.2 KMeans Clustering

Once the features are extracted, the next step is to cluster these descriptors into a visual vocabulary using the KMeans clustering algorithm. This is handled by the `Clusterer` class, which groups similar descriptors into K clusters. Each cluster centroid represents a "visual word," and the collection of all cluster centroids forms a visual vocabulary.

1. **Initialize KMeans**: KMeans is initialized with a specific number of clusters, which defines the size of the visual vocabulary.

2. **Combine Descriptors**: The descriptors from all the images are combined and fed into the KMeans algorithm to find K centroids.

3. **Fit the KMeans Model**: After fitting the KMeans model, each descriptor is assigned to the closest centroid.

4. **Create Histograms**: This mapping is used to create a histogram of the visual words for each image.

To determine the optimal number of clusters (K), the Elbow Method is used. This method involves plotting the inertia, which is the sum of squared distances from samples to their closest cluster center, against different values of K. As K increases, the inertia decreases. However, after a certain point, the reduction in inertia slows down, forming an "elbow" in the plot. The location of this elbow indicates the optimal number of clusters, as it represents a balance between the computational cost and the effectiveness of the clustering.

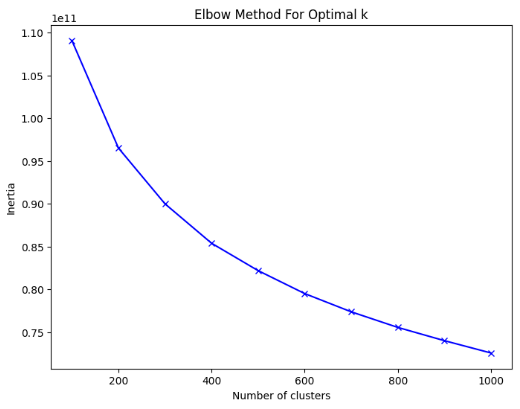


Figure 3.1: Elbow Method Plot

3.2.3 Visual Bag-of-Words Representation

The Visual Bag of Words model comes into play at this stage. Each image is represented as a histogram of visual words, where each bin corresponds to the number of occurrences of a particular visual word (cluster centroid) in the image. This histogram is a fixed-length vector, and all images are represented by vectors of the same length, regardless of the number of features or descriptors in each image.The transform method of the Clusterer class performs this step. For each image, it computes a histogram by counting how many of the image's descriptors belong to each cluster. This results in a vector of length K (the number of clusters), which is used as the image's feature representation.

3.2.4 Classifier

The final step in the pipeline is using the Support Vector Machine (SVM) classifier for image classification. The histograms of visual words created in the previous step serve as feature vectors, which are then passed to the SVM classifier. The SVM is trained to classify images based on their visual word histograms, enabling the model to predict the category of unseen images.

4 Results

4.1 Measuring the Accuracy of the Pipeline

4.1.1 Accuracy and Execution Time on Original Images

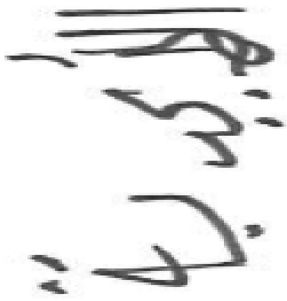
Model	Accuracy	Time
ORB	0.311234	1091.935860
SIFT	0.319214	450.765967

Table 4.1: Accuracy and execution time of the pipeline

The table presents the accuracy and execution time of the pipeline for two models, ORB and SIFT. In terms of accuracy, SIFT performs slightly better than ORB, with an accuracy of 31.9% compared to ORB's 31.1%. However, ORB takes significantly longer to execute, with a time of 1091 seconds, while SIFT executes much faster at 450 seconds. This indicates that while SIFT offers a marginal improvement in accuracy, ORB is much slower, highlighting the trade-off between computational efficiency and accuracy in this pipeline.

4.1.2 Accuracy on Modified Images

To evaluate the robustness of the pipeline to image transformations, the test set was modified using three operations: rotations, noise, and scaling. Images were rotated by specific angles such as 45°, 90°, and 135° to test orientation invariance. Gaussian noise was added to simulate random intensity variations, and scaling transformations were applied to assess the pipeline's performance under changes in image size. These modifications were designed to comprehensively evaluate the pipeline's resilience to various real-world conditions.


(a) Image Rotated by 90

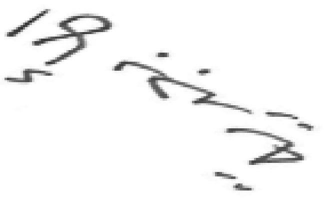

(b) Image Rotated by 135

Figure 4.1: Image Rotation

	Transformation	Accuracy (SIFT)	Accuracy (ORB)
0	Rotate 90	0.098600	0.060413
1	Rotate 45	0.042731	0.028733
2	Rotate 135	0.036960	0.031434

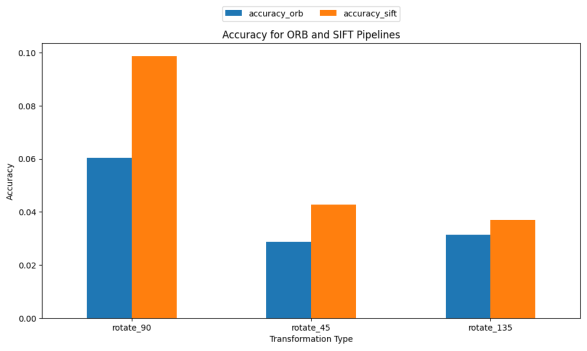


Figure 4.2: Accuracy of the pipeline for different rotations

The table highlights the pipeline's performance under image rotations (90°, 45°, and 135°) using SIFT and ORB. Both algorithms show reduced accuracy with increasing rotation. SIFT performs best at 90° (0.0986) but declines at 45° (0.0427) and 135° (0.0369). Similarly, ORB's accuracy is highest at 90° (0.0604) but decreases for 45° (0.0287) and 135° (0.0314). These results reflect the limitations of both algorithms in handling significant rotations, underscoring the need for enhanced robustness in feature extraction.

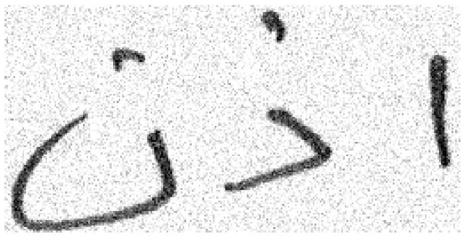


Figure 4.3: Gaussian Noisy Image

	Transformation	Accuracy (SIFT)	Accuracy (ORB)
0	noise_10	0.035241	0.070481
1	noise_20	0.040152	0.055869
2	noise_30	0.047520	0.063482

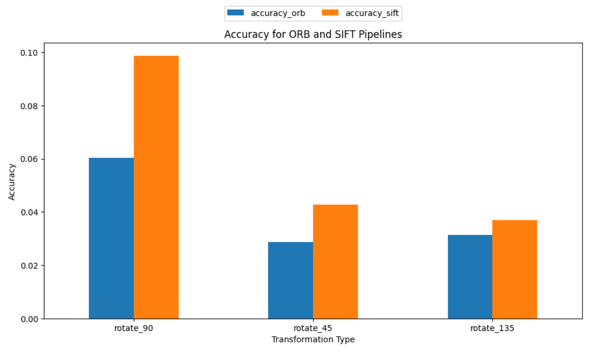


Figure 4.4: Accuracy of the pipeline for different noise levels

The table shows the accuracy of the pipeline under varying noise levels using SIFT and ORB. As the noise level increases from 10 to 30, both methods experience a decrease in accuracy, but ORB consistently outperforms SIFT at all noise levels. Specifically, ORB's accuracy is consistently higher, even as the noise level increases. This suggests that ORB is better at handling noise compared to SIFT in this context, potentially due to its efficiency and robustness in detecting features under challenging conditions. In contrast, SIFT seems to struggle more with higher noise levels, leading to lower accuracy.

	Transformation	Accuracy (SIFT)	Accuracy (ORB)
0	scale_0.5	0.169941	0.019524
1	scale_1.5	0.006262	0.017927
2	scale_0.75	0.276891	0.286100
3	scale_1.25	0.006017	0.051940

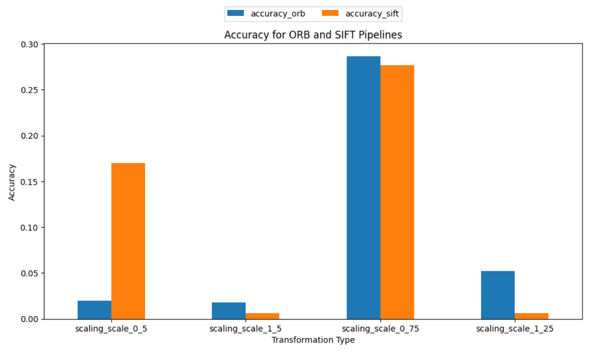


Figure 4.5: Accuracy of the pipeline for different scaling levels

The table presents the accuracy of the pipeline under varying scaling. The table shows the accuracy of SIFT and ORB under different scaling transformations. At scaling 0.5, SIFT outperforms ORB, but both methods struggle with larger scales (1.5). At 0.75, ORB slightly surpasses SIFT, while at 1.25, SIFT's accuracy drops drastically, and ORB performs better. This suggests that SIFT is more sensitive to extreme scaling, while ORB handles moderate scaling better but is less robust to very small or large scale changes.

4.1.3 Accuracy Across Different Cluster Sizes

Changing the cluster size affects model accuracy by balancing variability and precision. Smaller clusters may increase variability, making classification more difficult, while larger clusters can improve precision by better distinguishing data points. In this implementation, adjusting the cluster size helps find the optimal balance for improved classification performance.

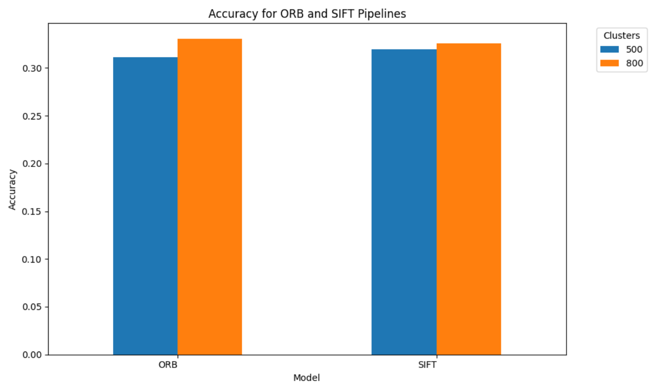


Figure 4.6: Accuracy Bar Plot for Different Cluster Sizes

4.2 Number of Keypoints Comparison

In the context of the Visual Bag of Words (BoW) pipeline for Arabic word classification, the number of keypoints detected by algorithms like SIFT and ORB plays a crucial role. A higher number of keypoints can improve the model's ability to capture important features from the images, which may enhance classification accuracy. However, this also increases the computational cost. Conversely, fewer keypoints can speed up the processing time but might result in a loss of important details, potentially affecting classification performance. Thus, understanding the impact of keypoint numbers helps in optimizing the balance between accuracy and efficiency in the Arabic word classification pipeline.

	ORB	SIFT
Average Keypoints	125.96	74.40
Sum Keypoints	1025832.00	605893.00

Table 4.2: Keypoints comparison between SIFT and ORB

The table shows the average number of keypoints detected by ORB and SIFT in the pipeline. On average, ORB detects more keypoints (125.96) compared to SIFT (74.40). This indicates that ORB may provide a more detailed representation of the image features, potentially capturing more information for the classification task. However, the higher number of keypoints in ORB could also increase computational complexity. Understanding this difference is important for optimizing the trade-off between classification accuracy and processing time in the pipeline.

5 Conclusion

This project evaluated the performance of local feature extraction techniques, Scale-Invariant Feature Transform (SIFT) and Oriented FAST and Rotated BRIEF (ORB), for Arabic handwritten word identification. The methodology involved preprocessing, feature extraction, clustering with KMeans, and classification using a Support Vector Machine (SVM). The key findings are summarized below:

- **Feature Extraction:**
  - **SIFT:**
    - Robust to transformations such as rotation and scaling, making it suitable for complex scenarios.
    - More sensitive to noise compared to ORB, leading to reduced performance in noisy environments.
  - **ORB:**
    - Faster and more computationally efficient than SIFT, making it ideal for real-time applications.
    - Exhibited greater noise resilience, consistently outperforming SIFT in noisy conditions.
    - Less effective under significant rotations and extreme scaling due to limited descriptor precision.
- **Clustering and Cluster Size:**
  - Larger cluster sizes created finer-grained visual vocabularies, improving classification accuracy by capturing subtle handwriting details.
  - Smaller cluster sizes oversimplified the feature space, reducing the ability to distinguish between handwriting styles.
  - The Elbow Method helped identify an optimal cluster size that balanced computational cost and classification performance.
- **Overall Insights:**
  - SIFT provided higher accuracy in transformation-heavy scenarios, such as rotations and scaling, but struggled with noise and required more computational resources.
  - ORB, while faster and more noise-resilient, performed less accurately in handling complex handwriting styles and transformations.
  - Larger cluster sizes significantly improved the pipeline's performance