Faculty of Engineering & Technology
Electrical & Computer Engineering Department
Computer Vision - ENCS5343

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

Prepared by: Hatem Hussein – 1200894
Nafe Abubaker – 1200047

Instructor: Dr. Aziz Qaroush

Section: 2

Date: December 23, 2024

Table of Contents

List of Figures

List of Tables

1. Introduction and Objective

In recent years, computer vision and pattern recognition have greatly improved our ability to analyze handwriting, including identifying who wrote a specific sample. This skill is useful in areas like authentication, forensic analysis, and personalized user tools. However, handwriting identification is challenging because of differences in writing styles, as well as variations caused by changes in size, orientation, lighting, and noise.

This study looks at two popular algorithms, **SIFT (Scale-Invariant Feature Transform)** and **ORB (Oriented FAST and Rotated BRIEF)**, to see how well they can identify handwriting authorship. These algorithms are tested using two methods: the **Visual Bag of Words (VBOW)** model and **K-Nearest Neighbors (KNN)-based feature matching**, with classification handled by a Support Vector Machine (SVM). We also test how these algorithms perform when handwriting samples are altered through scaling, rotation, changes in brightness, and added noise to mimic real-world conditions.

The goal of this study is to compare SIFT and ORB algorithms to see which performs better at identifying handwriting authorship. Specifically, we aim to:

1. **Measure Performance**:
    - Check accuracy as the percentage of handwriting samples correctly classified.
    - Record how long feature extraction and matching take.
2. **Test Robustness**:
    - See how well each method handles changes in size, rotation, lighting, and noise.
3. **Compare Keypoint Detection**:
    - Count the number of keypoints each algorithm detects and understand how this affects results.
4. **Provide Practical Insights**:
    - Highlight the trade-offs between accuracy, robustness, and speed for each algorithm to help decide which is better for real-world handwriting identification.

2. Background

Handwriting identification relies on recognizing unique patterns, such as stroke thickness, angle, and spacing, to determine authorship. This task can be influenced by various factors, including writing tools, paper texture, and environmental conditions, making robust algorithms essential.

2.1 SIFT (Scale-Invariant Feature Transform)

SIFT, introduced in 2004 by David Lowe, addresses scale variance in feature extraction. It identifies keypoints by approximating the Laplacian of Gaussian using the Difference of Gaussian (DoG) and detects local extrema. Keypoints are assigned orientations using histograms, ensuring robustness to rotation. Descriptors are generated by dividing a 16x16 neighborhood around the keypoint into 4x4 cells, producing a 128-dimensional feature vector.

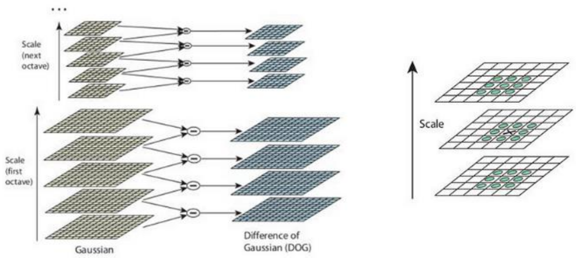

Figure 2.1: SIFT — Difference of Gaussian and Local Extrema Search

2.2 ORB (Oriented FAST and Rotated BRIEF)

ORB, introduced in 2011, combines the FAST keypoint detector and BRIEF descriptor for real-time applications. FAST detects keypoints by checking intensity differences in a circular neighborhood, while BRIEF generates compact binary descriptors. ORB enhances robustness by assigning orientations using intensity-weighted centroids.

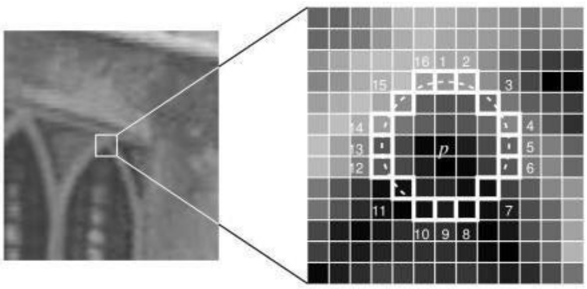

Figure 2.2: ORB — FAST key-point detection

2.3 Matching Methods

- **VBOW** creates histograms of visual words, summarizing keypoint distributions for classification.
- **KNN** matches descriptors to nearest counterparts, preserving spatial relationships.

3. Structure

The solution is structured into several key stages to ensure a systematic and efficient approach to handwriting identification. The primary stages are illustrated in the below figure:

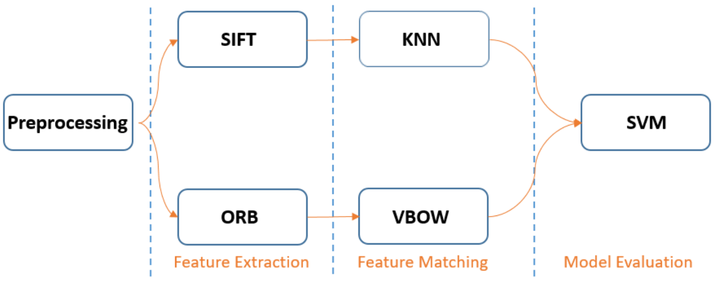

Figure 3.1: Final Structure

1. **Dataset Loading**:

    - Handwriting samples and their corresponding labels are loaded into the system. Labels represent the authorship of each handwriting sample.

2. **Preprocessing**:

    - The images are standardized to a consistent size and format. Basic preprocessing steps like grayscale conversion and noise reduction are applied to ensure uniformity and improve feature extraction performance.

3. **Feature Extraction**:

    - **SIFT** and **ORB** algorithms are applied to detect keypoints and compute descriptors for each image. The number of detected keypoints is also recorded for comparison.

4. **Feature Representation**:

    - For ORB, a **Visual Bag of Words (VBOW)** representation is created by clustering descriptors using K-Means and generating histograms.
    - For SIFT, **K-Nearest Neighbors (KNN)** is used to match descriptors, generating feature vectors based on match counts.

5. **Model Training and Testing**:

    - A Support Vector Machine (SVM) classifier is trained on the feature representations generated from both methods and evaluated on a separate dataset. Testing includes robustness analysis using transformed samples (scaled, rotated, brightened, and noisy) to assess accuracy under varying conditions.

4. Results and Analysis

Several classification models, including Random Forest (RF), Support Vector Machine (SVM), and Multi-Layer Perceptron (MLP), were evaluated for handwriting authorship identification. Among these, SVM demonstrated the highest accuracy, making it the most suitable choice for the final solution. The results justify its inclusion in both VBOW and KNN-based pipelines.

```
    Classifier  Accuracy (%)
0   SVM (SIFT)     30.691964
1    SVM (ORB)     16.741071
2    RF (SIFT)     19.196429
3     RF (ORB)      8.705357
4   MLP (SIFT)     26.339286
5    MLP (ORB)     10.714286
```

Figure 4.1: Trying several classification models

The four methods evaluated in this study (SIFT with VBOW, ORB with VBOW, SIFT with KNN, and ORB with KNN) demonstrate varying performance in terms of accuracy and time efficiency:

1. **Feature Extraction Time**:
    - SIFT is significantly slower in feature extraction compared to ORB, taking **33.59 seconds** for the training set and **9.68 seconds** for the testing set.
    - ORB, being designed for real-time applications, is much faster, completing extraction in **4.43 seconds** for training and **1.10 seconds** for testing.

```
Step 3: Feature Extraction
[Feature Extraction] SIFT extraction time: 33.59 seconds
[Feature Extraction] SIFT extraction time: 9.68 seconds
[Feature Extraction] ORB extraction time: 4.43 seconds
[Feature Extraction] ORB extraction time: 1.10 seconds
```

Figure 4.2: Feature extraction time

2. **Matching and Representation Time**:
    - The VBOW pipeline includes a time-intensive clustering step, taking over **600 seconds** for both SIFT and ORB. Histogram computation adds additional time, making VBOW relatively slow overall.
    - The KNN-based feature matching is computationally expensive during feature vector generation, especially for SIFT, which takes **2040.76 seconds**, compared to ORB's **497.97 seconds**.

```
Step 4a: Clustering for VBOW
[Clustering] VBOW clustering time: 626.45 seconds
[Clustering] VBOW clustering time: 619.73 seconds
[VBOW Computation] VBOW histogram computation time: 5.48 seconds
[VBOW Computation] VBOW histogram computation time: 1.12 seconds
[VBOW Computation] VBOW histogram computation time: 3.83 seconds
[VBOW Computation] VBOW histogram computation time: 0.96 seconds

---------------------------------------------------
Step 4b: KNN for Feature Matching
[KNN Matching] SIFT KNN training time: 0.02 seconds
[KNN Matching] ORB KNN training time: 0.00 seconds
[KNN Feature Vector Generation] Time: 2040.76 seconds
[KNN Feature Vector Generation] Time: 497.97 seconds
[KNN Feature Vector Generation] Time: 3433.51 seconds
[KNN Feature Vector Generation] Time: 863.51 seconds
```

Figure 4.3: Feature matching time

3. **Accuracy**:
    - **SIFT with KNN** achieves the highest accuracy at **36.50%**, demonstrating its effectiveness despite its computational cost.
    - **SIFT with VBOW** follows with an accuracy of **28.57%**, showing that VBOW is a viable alternative when accuracy is prioritized over time efficiency.
    - ORB underperforms in both pipelines, with **14.96% accuracy for VBOW** and **10.83% for KNN**, suggesting that while ORB is faster, it sacrifices precision in this task.

```
--------------------------------------------------
[Summary of Results]
SIFT VBOW Accuracy: 28.57%
ORB VBOW Accuracy: 14.96%
SIFT KNN Accuracy: 36.50%
ORB KNN Accuracy: 10.83%
```

Figure 4.4: Accuracy of different approaches

4. **Robustness:**

Following the robustness tests, SIFT with KNN and ORB with VBOW showed varying levels of resilience to transformations:

```
[Summary of Results After Transformations]
Scaled Transformation:
  SIFT KNN Accuracy: 35.27%
  ORB VBOW Accuracy: 10.27%
Rotated Transformation:
  SIFT KNN Accuracy: 35.94%
  ORB VBOW Accuracy: 14.96%
Brightened Transformation:
  SIFT KNN Accuracy: 30.80%
  ORB VBOW Accuracy: 10.16%
Noisy Transformation:
  SIFT KNN Accuracy: 12.95%
  ORB VBOW Accuracy: 4.91%
```

Figure 4.5: Robustness test output

SIFT KNN maintained higher accuracy under most transformations, especially for scaled and rotated images, proving its resilience. ORB VBOW struggled with all transformations, showing significant accuracy drops, particularly under noisy conditions.

5. **Key points**

ORB detected more keypoints on average (**107.98**) compared to SIFT (**94.33**) before transformations, suggesting better sensitivity to small features.

```
[Keypoint Detection] SIFT average keypoints: 94.33
```

Figure 4.6: SIFT average key points

```
[Keypoint Detection] ORB average keypoints: 107.98
```

Figure 4.7: ORB average key points

Overall, SIFT with KNN emerged as the most robust and accurate method, while ORB with VBOW offered a faster, albeit less precise, alternative for scenarios requiring lower computational cost.

5. Comparison and Visualization

The analysis highlights differences in performance across methods. **SIFT with KNN** is the most accurate, with 36.50% pre-transformation accuracy and strong robustness under scaling (35.27%) and rotation (35.94%), but it is computationally intensive. **SIFT with VBOW** is faster, offering moderate accuracy (28.57%) and robustness. **ORB-based methods**, while faster in extraction and matching, have lower accuracy and struggle under transformations like noise and brightness. ORB detects more keypoints, but this doesn't guarantee higher accuracy. Overall, SIFT methods excel in accuracy and robustness, while ORB methods are better for speed-sensitive tasks.The below table comparing all approaches have been used in order to reach the final solution:

Table 5.1: Comparison Table of Methods and Approaches

Method	Accuracy	Key Points (avg)	Robustness	Overall Performance
SIFT + KNN	High	Moderate	Excellent under scaling and rotation; struggles with noise and brightness.	Best accuracy overall, robust against most transformations, but computationally intensive for feature matching.
SIFT + VBOW	Moderate	Moderate	Decent robustness; better under scaling and rotation but less effective for noise and brightness.	Balanced approach with moderate accuracy and faster matching, suitable for scenarios where computational resources are limited.
ORB + KNN	Low	High	Poor robustness across all transformations; significant drop in performance under noise and brightness.	Faster matching but poor accuracy and robustness, making it less effective for handwriting identification tasks.
ORB + VBOW	Low	High	Limited robustness; struggles with noise and brightness; marginally better under scaling and rotation.	Fastest feature extraction and matching, but accuracy and robustness are suboptimal compared to SIFT methods.
SIFT + KNN (After Transformation)	Moderate to High	Moderate	Maintains robustness under scaling and rotation; struggles with noise and brightness.	Strong robustness and accuracy under certain transformations, making it the most reliable method for varied conditions.
ORB + VBOW (After Transformation)	Very Low	High	Poor robustness to all transformations; significant accuracy drop under noise and brightness.	While computationally efficient, it offers minimal accuracy and robustness, making it less ideal for real-world handwriting identification tasks.

The below figure comparing the accuracy for each combination:

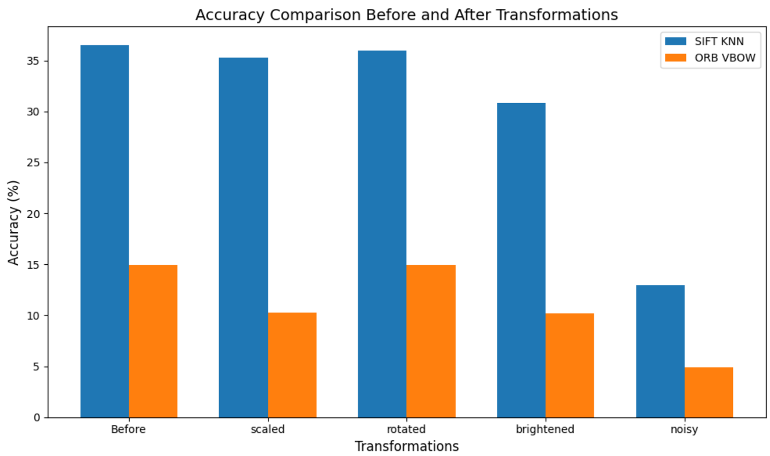

Figure 5.1: Accuracy Comparison Before and After Transformation

6. Conclusion

In conclusion, this study demonstrates the effectiveness of SIFT and ORB algorithms in handwriting authorship identification, emphasizing their suitability based on the dataset and task requirements. **SIFT with KNN** emerged as the most accurate and robust method, particularly under scaling and rotation transformations, making it ideal for datasets requiring high precision. **SIFT with VBOW** provided a good balance of speed and accuracy, suitable for resource-constrained scenarios. **ORB-based methods**, while computationally efficient, showed limitations in accuracy and robustness, especially under transformations like noise and brightness. Importantly, the choice of the approach depends heavily on the characteristics of the dataset. Datasets with more complex variations may benefit from robust methods like SIFT, whereas simpler datasets may allow for faster but less accurate methods like ORB. This underscores the importance of tailoring the approach to the specific dataset at hand.