Faculty of Engineering and Technology Electrical and Computer Engineering Department

Computer Vision

ENCS5343

Assignment NO. 2

"Arabic Handwritten Text Identification Using Local Feature Extraction Techniques"

Prepared by:

Nsralla Hasan 1200134

Dunia Jaser 1201345

Instructor:

Dr. Aziz Qaroush

Section NO.1

Birzeit

November – 2024

Abstract

In this project, the aim is to identify Arabic handwritten words using local feature extraction methods. The main goal is to have two popular feature descriptors, SIFT and ORB, implemented and compared to determine which one better captures the unique and consistent features of Arabic handwriting. These methods will be tested on the AHAWP dataset to ascertain which approach distinguishes between different handwritten words from various writers more accurately and efficiently. This work is important because it contributes to the improvement of the accuracy and reliability of automated handwritten text recognition systems. Due to the great variations in letter shapes, spacing, and writing styles of Arabic handwriting, it is challenging to recognize it accurately. By focusing on effective feature extraction, matching, and classification, our project enhances computer vision systems' ability to process real-world handwritten Arabic documents.

Introduction

Arabic handwritten text identification is challenging due to the wide variety of handwriting styles. The shapes of Arabic letters can change depending on the writer, the size of the text, and how the letters are connected. These differences create difficulties for computers to accurately recognize handwritten words.

The goal of this assignment is to identify Arabic handwritten words using advanced methods like SIFT and ORB. These techniques detect key points in the text images that remain consistent, even if the text changes in size, angle, or lighting.

Scale-Invariant Feature Transform (SIFT)

The core objective of SIFT is to detect distinctive and invariant keypoints in images and generate robust descriptors that can be used to match those keypoints across different images. The algorithm achieves scale invariance, meaning it can detect the same features regardless of the image's scale. Additionally, it is rotation invariant, which means the keypoints detected remain consistent even if the image is rotated.

How The SIFT Algorithm Works?

The SIFT algorithm is designed to detect distinctive features. These keypoints are unique areas that can be used to recognize objects in different images, even if the image is scaled, rotated, or slightly changed. Following are the main steps in the SIFT algorithm:

1) Scale-space Extrema Detection
2) Keypoint Localization
3) Orientation Assignment
4) Keypoint Descriptor

Oriented FAST and Rotated BRIEF (ORB)

ORB is a robust alternative to patented methods like SIFT and SURF. It combines the FAST keypoint detector and BRIEF descriptor but improves upon them for better performance. ORB applies FAST to detect keypoints and refines them using the Harris corner measure to ensure only the most prominent are selected. It addresses FAST's lack of rotational invariance by calculating the orientation of keypoints based on the intensity-weighted centroid of their patches.

The descriptor is enhanced by rotating the standard BRIEF descriptor according to the keypoint orientation, a process termed "steering." This adjustment ensures the descriptor is rotation-invariant and remains effective across different viewing conditions. ORB also selects the most discriminative binary tests for its descriptor through a greedy algorithm, optimizing for high variance and minimal correlation to maximize the descriptor's robustness.

For efficient descriptor matching, ORB uses an advanced version of Locality Sensitive Hashing, making it faster and more suitable for real-time applications and low-power devices like those used in panorama stitching.

Structure

Dataset Preparation

The AHAWP (Arabic Handwritten Automatic Word Processing) dataset, designed for recognizing handwritten Arabic text, was utilized. It contains word-level data with ten distinct Arabic words, handwritten by 82 participants who each provided ten samples, resulting in 8,144-word images. Each participant is anonymously identified by a user ID to ensure privacy while enabling sample tracking.

The dataset was split into training (80%) and testing (20%) subsets. Word images were named with structured filenames indicating the writer (user ID), word (word ID), and image index. To ensure balanced representation, data was grouped by word ID, and each group was split into training and testing subsets. Each image went through preprocessing, including conversion to grayscale to reduce computational complexity. The training data developed feature extraction and classification models, while the testing data evaluated their performance.

Two systems were built: one measuring the percentage of correctly matched key points using Cosine similarity and another using an SVM classifier to identify the writer of each word in the testing set.

Image Augmentation

To enhance dataset diversity and model robustness, augmentation techniques were applied, including scaling to simulate size variations, rotation for perspective changes, illumination adjustments for varying lighting, and noise addition (Gaussian and salt-and-pepper) to mimic sensor and environmental distortions.

Feature Extraction

Features were extracted using the ORB (Oriented FAST and Rotated BRIEF) algorithm for its speed and rotation handling. SIFT (Scale-Invariant Feature Transform) was also used for comparison due to its effectiveness with scale and rotation variations.

Visual Vocabulary Construction

The Bag-of-Visual-Words (BoVW) approach was used to construct a visual vocabulary. Descriptors were clustered using KMeans, and experimentation with various cluster sizes (100, 500, 700, and 1100) identified 900 as the optimal balance between accuracy and computational efficiency.

Image Representation

Each image was represented as a histogram of visual words, quantifying the frequency of each visual word in the image. This histogram was normalized to ensure comparability across images and weighted using Term Frequency-Inverse Document Frequency (TF-IDF) to emphasize distinctive features while down-weighting commonly occurring ones.

Feature Scaling

To standardize the feature space, a StandardScaler was applied.

Classification Model

SVM

A Support Vector Machine (SVM) with an RBF kernel was selected for its effectiveness with high-dimensional data. Various configurations were tested, including linear and RBF kernels and different C values (1, 10, 100). The final model used an RBF kernel, C=10, and auto gamma, trained on scaled features from the training images for consistency and improved performance.

Random Forest

A Random Forest Classifier was chosen for its robustness and ability to handle high-dimensional data. Using GridSearchCV, optimal parameters were identified: max_depth = None, min_samples_leaf=2, min_samples_split = 2, and n_estimators = 200. This configuration balanced model complexity and accuracy. Note: SVM outperformed Random Forest and was selected for key-point matching due to its superior accuracy and efficiency.

Testing and Evaluation

Testing images underwent the same preprocessing, augmentation, and feature extraction as the training data. Visual word histograms were converted to TF-IDF representations and scaled using the training scaler. The SVM classifier predicted testing labels, and accuracy was used to evaluate performance.

Key-points Matching

The pipeline was similar to the SVM process, except the focus was on matching key-points between query images and training data based on feature similarities to ensure accurate correspondence. The process went as following:

a. Feature Descriptor Comparison
    - Descriptor Representation:
      Each key-point detected by ORB or SIFT is paired with a descriptor.
    - Distance Metrics:
      Cosine similarity, which measures the cosine of the angle between descriptor vectors, is used for comparison. Higher similarity indicates closer matches.

b. Matching Algorithms
    - FLANN-Based Matcher:
      The Fast Library for Approximate Nearest Neighbors (FLANN) matcher efficiently handles high-dimensional data, particularly with binary descriptors from ORB or SIFT. It uses approximate nearest neighbors to speed up matching with minimal accuracy loss.
    - Locality-Sensitive Hashing (LSH):
      To boost FLANN's performance, LSH hashes descriptors into buckets, ensuring similar descriptors are grouped together for faster matching.

c. Lowe's Ratio Test
   Lowe's ratio test reduces false matches by comparing the distances of the closest and second-closest neighbors for each descriptor. Matches are considered reliable if the closest distance is less than 70% of the second-closest distance.

d. Good Matches Ratio Calculation

- Good Matches Identification:
  "Good matches" are those passing Lowe's ratio test. The ratio of good matches to total matches attempted is calculated, indicating the reliability of key-point correspondences between query and training images.

Results and visualizations

Comparison of SIFT and ORB Algorithms

The performance of SIFT and ORB algorithms was evaluated for Arabic handwritten text identification on the AHAWP dataset, focusing on key points detected, accuracy, feature representation, and computational efficiency.

Key Points Detected

SIFT detected 5,818,279 key points in the training dataset and 1,440,823 in testing, whereas ORB detected significantly more, with 12,397,036 in training and 3,073,723 in testing. Despite detecting more key points, ORB's performance did not correlate directly with this advantage.

Accuracy

SIFT achieved 47.78% accuracy with SVM and 31.21% with Random Forest, outperforming ORB, which scored 28.20% and 14.57%, respectively. SIFT's superior descriptors likely account for its robustness against variations in scale, rotation, and illumination.

Feature Representation

The Bag-of-Visual-Words visual vocabulary contained 900 clusters. SIFT used 128-dimensional descriptors, providing richer information, while ORB's 32-dimensional descriptors offered computational efficiency but reduced accuracy due to less detailed feature representation.

Computational Efficiency

ORB was faster during feature extraction, taking 39.96 seconds compared to SIFT's 21.03 seconds, and detected more key points. However, SIFT proved more reliable, particularly in challenging imaging conditions. Matching times were 105 seconds for SIFT and 119.90 seconds for ORB, emphasizing a trade-off between speed and reliability.

Observation on using SIFT, ORB in the classifiers

- **SIFT Strengths:** SIFT excels in accuracy and robustness. Its ability to produce high-dimensional descriptors enables better differentiation between key points.

- **ORB Strengths:** ORB is faster and detects a higher number of key points. It is better suited for applications where computational resources are limited or real-time processing is required. However, the lower descriptor dimensionality compromises its performance in tasks requiring high accuracy and robustness.

- **Trade-Offs:** The results highlight the trade-off between computational efficiency and classification performance. While ORB offers efficiency, SIFT's higher accuracy makes it a better choice for this specific task.

SIFT

The results demonstrate the application of the SIFT algorithm, showcasing detected keypoints on query and training images. The visualizations illustrate the effects of scaling, rotation, illumination changes, and noise on the query images, emphasizing the robustness of SIFT's keypoint detection under these transformations.

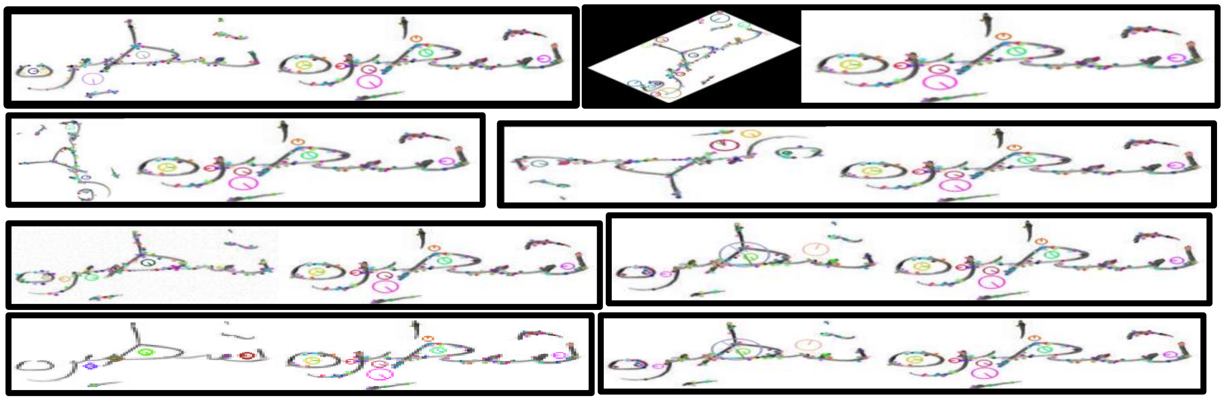


Figure 1: Visualization of SIFT Key Point Detection on Query and Training Images

The table below presents the robustness test results for some query images.

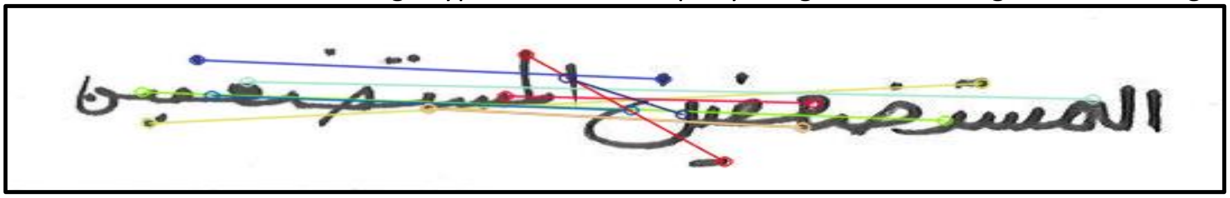
Table 1: Impact of Transformations on SIFT Descriptor Similarity for a Query Image

Test	Value
Original Similarity	0.2621
Scale 0.5	Similarity = 0.2538
Scale 1.5	Similarity = 0.2656
Rotation 45°	Similarity = 0.2299
Rotation 90°	Similarity = 0.2286
Rotation 180°	Similarity = 0.2148
Noise var=50	Similarity = 0.2049
Illumination gamma=0.5	Similarity = 0.2050
Illumination gamma=1.5	Similarity = 0.2567

Table 2: Detailed Robustness Evaluation of SIFT on Query Image 'user051_ghazaal_009' with Various Transformations

Test	Value
Robustness Test for Query	testing_images\user051_ghazaal_009.png
Best Match Training Image	training_images\user051_ghazaal_003.png
Original Similarity	0.3268
Scale 0.5	Similarity = 0.2865
Scale 1.5	Similarity = 0.1839
Rotation 45°	Similarity = 0.1860
Rotation 90°	Similarity = 0.2906
Rotation 180°	Similarity = 0.2776
Noise var=50	Similarity = 0.3499
Illumination gamma=0.5	Similarity = 0.4213
Illumination gamma=1.5	Similarity = 0.2825

Some of the results of matching keypoints between query images and training database images.



ORB

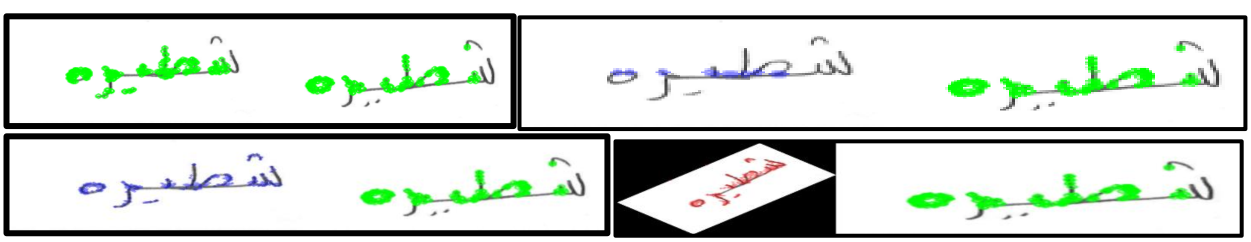
The following are some results of retrieved images by query image.

Table 3: ORB Robustness Test Results for Query Image 'user045_abjadiyah_039' Under Various Conditions

Test	Value
Robustness Test for Query	testing_images\user045_abjadiyah_039.png
Best Match Training Image	training_images\user044_abjadiyah_036.png
Original Similarity	0.2976
Scale 0.5	Similarity = 0.0357
Scale 1.5	Similarity = 0.1839
Rotation 45°	Similarity = 0.2397
Rotation 90°	Similarity = 0.2757
Rotation 180°	Similarity = 0.2771
Noise var=50	Similarity = 0.1654
Illumination gamma=0.5	Similarity = 0.2301
Illumination gamma=1.5	Similarity = 0.2380

Table 4: Detailed Robustness Analysis for ORB on Query Image 'user025_qashtah_022' Across Diverse Conditions

Test	Value
Robustness Test for Query	testing_images\user025_qashtah_022.png
Best Match Training Image	training_images\user079_qashtah_030.png
Original Similarity	0.3162
Scale 0.5	Similarity = 0.0607
Scale 1.5	Similarity = 0.2738
Rotation 45°	Similarity = 0.2866
Rotation 90°	Similarity = 0.3149
Rotation 180°	Similarity = 0.3304
Noise var=50	Similarity = 0.2717
Illumination gamma=0.5	Similarity = 0.3028
Illumination gamma=1.5	Similarity = 0.2861



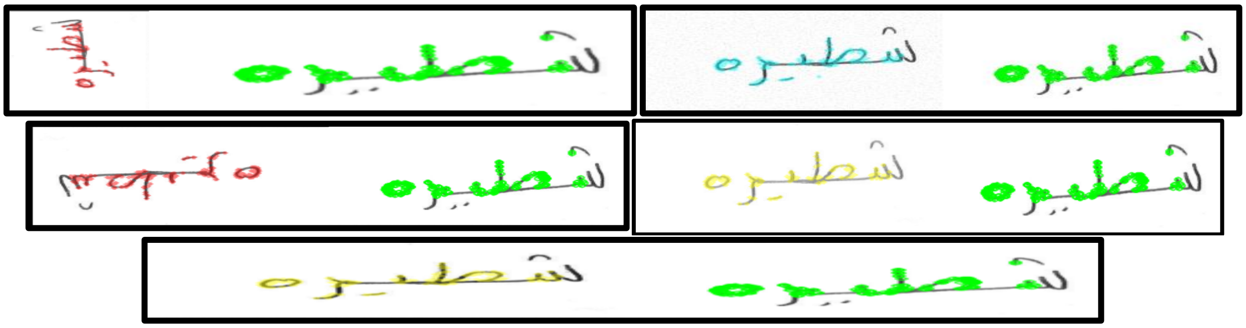


Figure 2: Visualization of ORB Key Point Detection on Query and Training Images

Observations on the performance of the algorithms (SIFT and ORB) on Matching key points

The following table highlights the performance of SIFT and ORB algorithms under various conditions, emphasizing their strengths and weaknesses:

Table 5: Comparison of SIFT and ORB Algorithms Across Different Conditions

Condition	SIFT	ORB	Better Algorithm
Original Similarity	Performs well but generally slightly lower	Typically higher similarity scores.	ORB
Scale Variations	Strongly robust to smaller scales.	Struggles with smaller scales; improves at larger scales.	SIFT (Overall)
Rotation	Performs well at smaller rotations.	More robust at higher rotations,especially 180	ORB (Rotation)
Noise	Handles noise effectively and consistently.	Less robust to noise; scores are lower.	SIFT
Illumination	Consistent under varying lighting conditions.	Occasionally matches SIFT under brighter illumination.	SIFT (Overall)

Both SIFT and ORB demonstrate distinct strengths that make them suitable for specific scenarios in Arabic handwritten text identification. SIFT excels in handling challenging conditions such as noise, varying illumination, and small-scale variations, ensuring consistent and robust performance. In contrast, ORB outperforms SIFT in scenarios involving large rotations (e.g., 180°) and often achieves higher similarity scores in original matches. While SIFT offers better reliability and versatility under diverse conditions, ORB provides a faster and more efficient alternative, making it preferable for applications prioritizing computational efficiency. The optimal choice between the two depends on the task requirements and environmental constraints.