Report

Biometric Template Protection using Fuzzy Vault

~ *Sagnik Chakraborty* [22101106020]

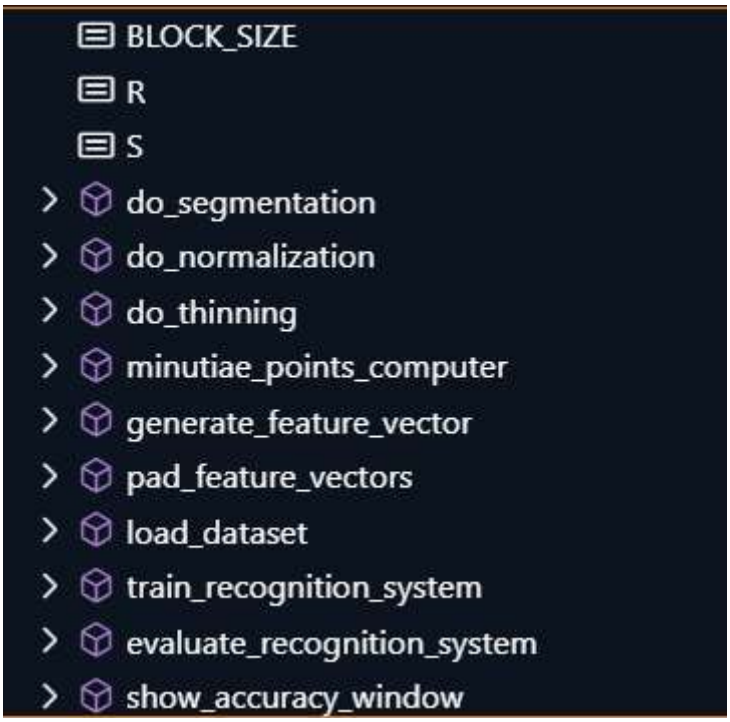
Introduction

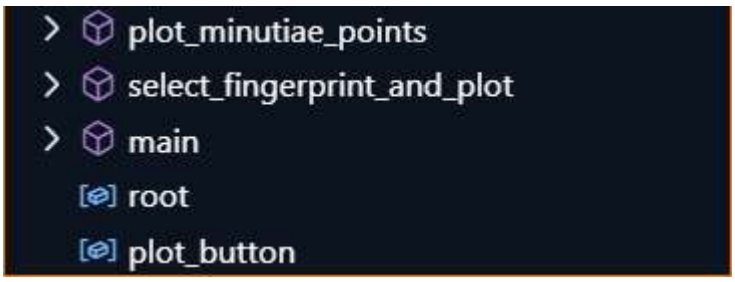
The fuzzy vault is a cryptographic technique designed to secure biometric templates such as fingerprints and palmprints. These templates are unique to individuals and provide a reliable means of authentication. However, protecting this sensitive biometric data from unauthorized access is crucial. The fuzzy vault provides a mechanism to encode biometric features with a secret key, ensuring that only authorized users can access the stored information. The fuzzy vault is built on principles of error-correcting codes, where the biometric data is encoded into a vault using a polynomial equation. By adding false or "chaff" points, the genuine biometric points are hidden, enhancing security. The vault can only be decoded with the correct key, ensuring that sensitive biometric information is protected from adversaries.

Key Features

1. **Preprocessing**:
    a. Segmentation, normalization, and thinning of fingerprint images.
2. **Feature Extraction**:
    a. Minutiae points are extracted from thinned images.
    b. These points are transformed into feature vectors for classification.
3. **Training and Recognition**:
    a. Support Vector Machines (SVMs) are trained on extracted features.
    b. Grid search is employed for hyperparameter tuning.
4. **Evaluation**:
    a. Accuracy is calculated to evaluate the performance of the trained model.
5. **Graphical User Interface**:
    a. A simple GUI using tkinter enables fingerprint selection and minutiae plotting.

Methodology





1. **Preprocessing**
    a. **Segmentation**: Non-informative regions of the fingerprint are identified using block-wise variance and removed.
    b. **Normalization**: Image intensity is adjusted to ensure consistent feature extraction.
    c. **Thinning**: Skeletonization of the binary fingerprint image to simplify feature extraction.

2. **Minutiae Extraction**
    a. Minutiae points, representing ridge endings and bifurcations, are detected from the skeletonized image.
    b. These points are stored with their coordinates and orientations.

3. **Feature Vector Generation**
    a. Extracted minutiae points are transformed into fixed-size feature vectors for classification.
    b. Padding or truncation ensures uniform vector lengths across all samples.

4. **Dataset Preparation**
    a. Datasets are structured with subdirectories representing different fingerprint categories.

b. Feature vectors are generated for training and testing images.

5. **Training**

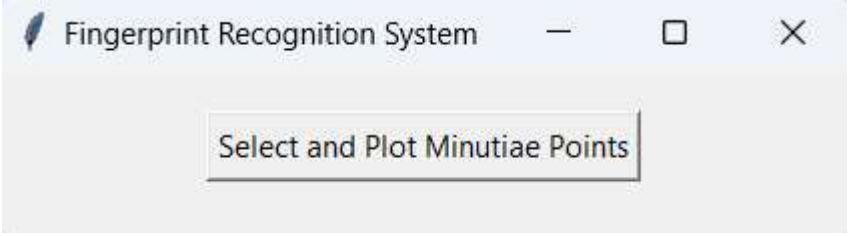
   a. A Support Vector Machine (SVM) classifier is trained on the feature vectors.
   b. GridSearchCV is used for tuning hyperparameters like `C`, `kernel`, and `gamma`.

6. **Evaluation**

   a. Recognition accuracy is computed on the test dataset using the trained model.
   b. Results are displayed through a GUI window.

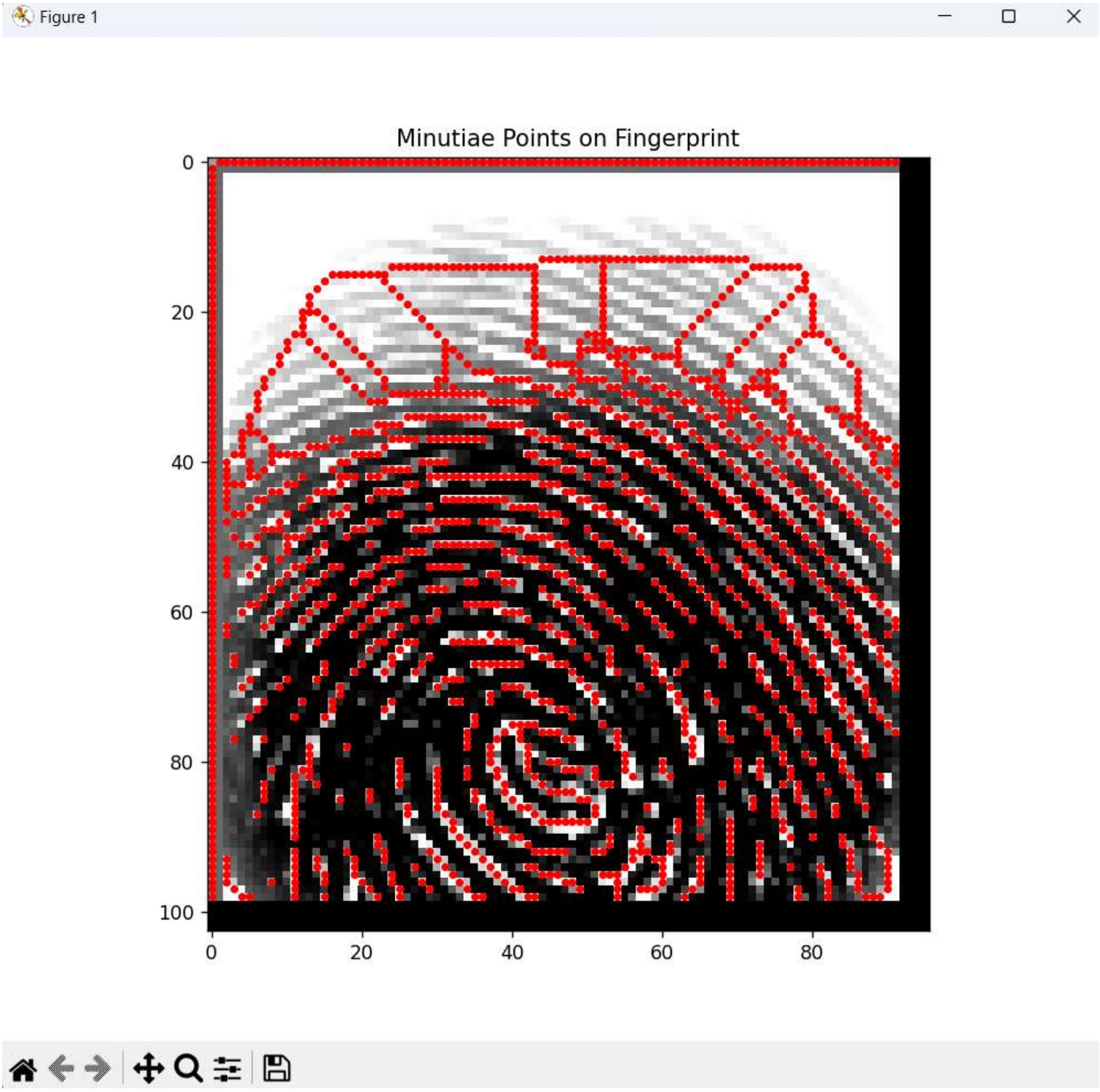
7. **Graphical Interface**

   a. Users can interact with the system to select fingerprint images and visualize minutiae points.



Results

- **Segmentation and Thinning**:
  - Successfully preprocesses fingerprint images to a skeletonized form for minutiae extraction.
- **Feature Representation**:
  - Minutiae points are accurately represented as feature vectors.



- **Classification Accuracy**:
  - Recognition accuracy is computed and displayed using a GUI.

```
Processing image: fingerprint_dataset2\SOCOFing\Altered\Altered-Easy\4__M\4__M_Right_index_finger_Zcut.BMP
Processing image: fingerprint_dataset2\SOCOFing\Altered\Altered-Easy\4__M\4__M_Right_little_finger_CR.BMP
Processing image: fingerprint_dataset2\SOCOFing\Altered\Altered-Easy\4__M\4__M_Right_little_finger_Obl.BMP
Processing image: fingerprint_dataset2\SOCOFing\Altered\Altered-Easy\4__M\4__M_Right_little_finger_Zcut.BMP
Processing image: fingerprint_dataset2\SOCOFing\Altered\Altered-Easy\4__M\4__M_Right_middle_finger_CR.BMP
Processing image: fingerprint_dataset2\SOCOFing\Altered\Altered-Easy\4__M\4__M_Right_middle_finger_Obl.BMP
Processing image: fingerprint_dataset2\SOCOFing\Altered\Altered-Easy\4__M\4__M_Right_middle_finger_Zcut.BMP
Processing image: fingerprint_dataset2\SOCOFing\Altered\Altered-Easy\4__M\4__M_Right_ring_finger_CR.BMP
Processing image: fingerprint_dataset2\SOCOFing\Altered\Altered-Easy\4__M\4__M_Right_ring_finger_Obl.BMP
Processing image: fingerprint_dataset2\SOCOFing\Altered\Altered-Easy\4__M\4__M_Right_ring_finger_Zcut.BMP
Processing image: fingerprint_dataset2\SOCOFing\Altered\Altered-Easy\4__M\4__M_Right_thumb_finger_CR.BMP
Processing image: fingerprint_dataset2\SOCOFing\Altered\Altered-Easy\4__M\4__M_Right_thumb_finger_Obl.BMP
Processing image: fingerprint_dataset2\SOCOFing\Altered\Altered-Easy\4__M\4__M_Right_thumb_finger_Zcut.BMP
Normalizing features...
Training fingerprint recognition system...
Best Parameters: {'C': 10, 'gamma': 'auto', 'kernel': 'rbf'}
Evaluating fingerprint recognition system...
Fingerprint Recognition Accuracy: 87.50%
```

Limitations

1. Variability in fingerprint quality due to noise, smudges, or incomplete ridges may affect performance.
2. The fixed-length feature vector approach may truncate meaningful minutiae in complex fingerprints.
3. SVM classifiers, while robust, may not perform well on highly imbalanced datasets without additional preprocessing.

Conclusion

The fingerprint recognition system demonstrates the ability to preprocess, extract features, and

classify fingerprints effectively. While the results show promising accuracy, further improvements, such as advanced deep learning models or adaptive feature extraction, could enhance robustness and scalability.

Future Work

1. **Deep Learning**:
    a. Employ CNNs or other architectures for automatic feature extraction.
2. **Improved Preprocessing**:
    a. Enhance segmentation and normalization to handle noisy or partial fingerprints better.
3. **Cross-Matching**:
    a. Implement cross-matching capabilities to compare fingerprints across databases.

Usage Instructions

1. Run the script to launch the GUI.
2. Use the "Select and Plot Minutiae Points" button to upload a fingerprint and visualize its features.
3. The system will automatically train and evaluate the recognition model.

References

1. K. Nandakumar, A. K. Jain, and S. Pankanti, “Fingerprint-based Fuzzy Vault: Implementation and Performance,” *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 4, pp. 744–757, 2007.
2. A. Juels and M. Sudan, “A Fuzzy Vault Scheme,” *Proceedings of IEEE International Symposium on Inxformation Theory*, vol. 6, no. 3, pp. 408, 2002.
3. K. Nandakumar, “Multibiometric System: Fusion Strategies and Template Security,” *PhD Thesis*, Michigan State University, January 2008.
4. V. Evelyn Brindha, “Biometric Template Security using Fuzzy Vault,” *IEEE 15th International Symposium on Consumer Electronics*, 2011.

This report highlights the methodology, achievements, and areas for improvement in the fingerprint recognition project. For implementation, refer to the attached source code.