JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

An Innovative Template Solution for Securing Authentication in Resource-Constrained IoT Environments

K. Rachana*[1], B. Manasa[2], U. Bhavana[3] , G. Shekar [5]
[*1] UG Student, Department of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad, Telangana,India
[2] UG Student, Department of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad, Telangana,India
[3] UG Student, Department of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad, Telangana,India.
[4]Assistant Professor, Department of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad, Telangana,India.

Abstract: This script facilitates fingerprint authentication via the SIFT algorithm. Users pick a fingerprint image, which the script then compares against a collection of stored fingerprint images. Using SIFT, it identifies and describes unique features (keypoints) in both the input and stored images. For each stored fingerprint, the script computes matches between the input and stored keypoints. If enough matches are found, indicating significant similarity, the script proceeds to perform homography and displays the matching keypoints. Successful matches grant access. FLANN is used for efficient keypoint matching and homography calculation. Access is granted upon successful match; otherwise, an error message indicates unauthorized access. This serves as a foundational example for biometric security and access control, with potential for further development.

IndexTerms **- P Biometric Security, Access Control, Image Matching, Feature Descriptor, Error Handling**

I. INTRODUCTION

Fingerprint recognition is a widely adopted biometric authentication method known for its reliability. This Python script harnesses the power of the SIFT algorithm to execute fingerprint authentication. SIFT excels at identifying and aligning distinct features in images, making it an ideal choice for tasks like fingerprint recognition.[1] The user initiates the process by selecting an input fingerprint image. The script then employs SIFT to extract key features from this image. These features are subsequently compared against those extracted from a database of stored fingerprint images. Upon discovering a substantial number of matches, indicating significant similarity, the script proceeds to perform homography and presents a visualization of the matching keypoints. It's crucial to ensure these programs remain fair and unbiased, avoiding wrongful punishment or censorship. Collaborating to address these challenges can foster a more pleasant social media environment for all users.

I.I Objective

[2] The provided Python script aims to create a fingerprint authentication system utilizing the Scale-Invariant Feature Transform (SIFT) algorithm. The project emphasizes the utilization of unique fingerprint features for robust identity verification. By employing the SIFT algorithm, the script identifies and describes distinct features in both the input fingerprint image and a collection of stored fingerprint images. Key features are efficiently matched using the Fast Library for Approximate Nearest Neighbors (FLANN) algorithm. Successful matches prompt the calculation of homography to align matching keypoints, visually representing similarities. The project features a user-friendly graphical interface enabling users to select an input fingerprint image and receive authentication feedback. The primary objective is to provide a foundational example of biometric security, encouraging exploration and understanding of fingerprint recognition for identity verification in Python.

II. EXISTING SYSTEMS

PCA-Based Systems:

Algorithm: Principal Component Analysis (PCA) has been applied to fingerprint recognition, where the principal components of fingerprint images are used for feature extraction and matching.[3]

Texture-Based Systems:

Algorithm: These systems analyze the textural patterns of fingerprints. Local Binary Pattern (LBP) is a popular texture-based algorithm that describes the microstructure of fingerprint ridges and furrows.

PROPOSED SYSTEM

The Python script for fingerprint authentication using the SIFT algorithm stands as a standalone educational example. It doesn't indicate integration with any pre-existing fingerprint recognition system or specialized libraries tailored for biometric [4]authentication. Rather, the script utilizes the OpenCV library, which encompasses implementations of computer vision algorithms such as SIFT, FLANN, and homography. It's crucial to recognize that real-world fingerprint recognition systems typically encompass more extensive frameworks and libraries explicitly designed for biometric authentication.

III. METHODOLOGIES

[5] The project presents a Flask-based web application designed for converting audio to text, offering users a seamless tool for transcribing spoken words into written form. Using the SpeechRecognition library, the application enables users to upload audio files through an intuitive web interface. Its main functionality relies on leveraging the Google Speech Recognition API to accurately transcribe the audio content. Adding to its user-friendly interface, IPython's Audio module facilitates playback of uploaded audio files directly on the web platform. The project aims to provide an efficient and accessible solution for individuals seeking speech-to-text conversion, with clear pathways for managing files and displaying recognition results on the web page. Despite its title,[6] "Audio to Text Generation with Created Dataset," the code primarily focuses on implementing speech recognition from user-uploaded audio files. Detailed information regarding the creation and use of a custom dataset is not explicitly addressed in the provided code snippet.

1.NumPy (numpy):

NumPy is a fundamental package for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these arrays.

2. OpenCV (cv2):

OpenCV is an open-source computer vision and image processing library. It provides tools for image and video analysis, including various algorithms for feature detection, matching, and image manipulation.

3. Glob (glob):

The glob module is used to retrieve file paths that match a specified pattern (in this case, all PNG images in the current directory).

4. Tkinter (tkinter):

Tkinter is Python's standard GUI (Graphical User Interface) toolkit. It is used to create the graphical interface for the user to select an input fingerprint image.

REQUIREMENTS ENGINEERING

[12]. These are the requirements for doing the project. Without using these tools & software's we can't do the project. So we have two requirements to do the project. They are

- Hardware Requirements.
- Software Requirements

1.HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system do and not how it should be implemented.[7]

- PROCESSOR : Pentium i3 Processor
- RAM : 2GB DD RAM
- HARD DISK : 250 GB

2. SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.[8]

- BACK END : PYTHON
- OPERATING SYSTEM : WINDOWS 7
- IDE : Spyder3

3. FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior, This paper has explained about a library named as Sparx which provides the solution for a programmer to acquire cleaned data for further analysis. Sparx is an exclusive data-preprocessing library, which involves transforming raw data into a machine-understandable format. The intention behind developing Sparx was to build a better, automated data-preprocessing library.

4. NON-FUNCTIONAL REQUIREMENTS

The major non-functional Requirements of the system are as follows

Usability :

The system is designed with completely automated process hence there is no or less user intervention.

Reliability :

The system is more reliable because of the qualities that are inherited from the chosen platform java. The code built by using python is more reliable.

Performance :

This system is developing in the high level languages and using the advanced front-end and back-end technologies it will give response to the end user on client system with in very less time.

Supportability :

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is built into the system.

Implementation :

The system is implemented in web environment using Django framework. The server is used as the web server and windows xp professional is used as the platform. Interface the user interface is based on Django provides web application.

IV. SYSTEM ARCHITECTURE

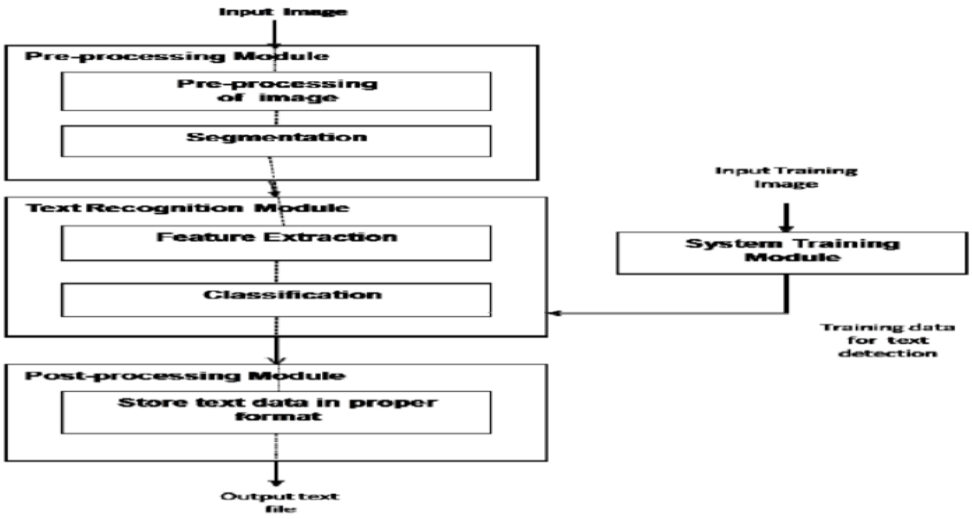

FIG: System Architecture Model

V. LITERATURE SURVEY

[9]The Internet of Things (IoT) is the next era of communication. Using the IoT, physical objects can be empowered to create, receive, and exchange data in a seamless manner. Various IoT applications focus on automating different tasks and are trying to empower the inanimate physical objects to act without any human intervention. The existing and upcoming IoT applications are highly promising to increase the level of comfort, efficiency, and automation for the users. To be able to implement such a world in an ever-growing fashion requires high security, privacy, authentication, and recovery from attacks. In this regard, it is imperative to make the required changes in the architecture of the IoT applications for achieving end-to-end secure IoT environments. In this paper, a detailed review of the security-related challenges and sources of threat in the IoT applications is presented. After discussing the security issues, various emerging and existing technologies focused on achieving a high degree of trust in the IoT applications are discussed. Four different technologies, blockchain, fog computing, edge computing, and machine learning, to increase the level of security in IoT are discussed.

On-demand ride and ride-sharing services have revolutionized the point-to-point transportation market and they are rapidly gaining acceptance among customers worldwide. Alone, Uber and Lyft are providing over 11 million rides per day (DMR, 2018a,b). These services are provided using a client-server infrastructure. The client is a Smartphone-based application used for: (i) registering riders and drivers, (ii) connecting drivers with riders, (iii) car-sharing to share the expenses, minimize traffic congestion and saving traveling time, (iv) allowing customers to book their rides. The server typically, run by multi-national companies such as Uber, Ola, Lyft, BlaBlaCar, manages drivers and customers registrations, allocates ride-assignments, sets tariffs, guarantees payments, ensures safety and security of riders, etc.[10] However, the reliability of drivers have emerged as a critical problem, and as a consequence, issues related to riders safety and security have started surfacing. The lack of robust driver verification mechanisms has opened a room to an increasing number of misconducts (i.e., drivers subcontracting ride- assignments to an unauthorized person, registered drivers sharing their registration with other people whose eligibility to drive is not justified.

SNAPSHOTS

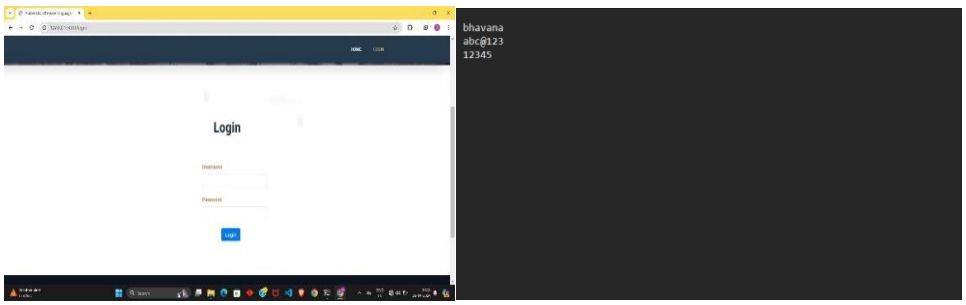

FIG : User Information **FIG: Input**

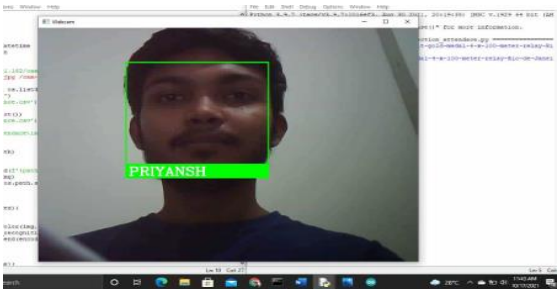

FIG : Result

VI. CONCLUSION

In summary, the script lays the groundwork for a fingerprint authentication system utilizing the Scale-Invariant Feature Transform (SIFT) algorithm. By efficiently computing keypoints and their matches using the FLANN algorithm, it enables a dependable comparison between the input fingerprint and a repository of stored fingerprints, thus facilitating secure access control. However, there are avenues for future improvement. [11]Firstly, enhancing the system's robustness and accuracy could involve integrating advanced feature matching techniques or exploring alternative cutting-edge algorithms beyond SIFT. Additionally, incorporating machine learning approaches could lead to more intelligent decision-making in the authentication process. Scalability could be enhanced by expanding the database of stored fingerprints and optimizing matching criteria for improved accuracy. Implementing additional security measures, such as encrypting stored fingerprints and regularly updating to counter emerging threats, would bolster the system's resilience. Furthermore, exploring integration with other biometric modalities or multi-factor authentication methods could enhance its effectiveness in various security scenarios. Continuous research and development efforts are crucial for refining and expanding the capabilities of this system for broader applications in biometric security and access control.

VII. REFERENCES

[1] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "Asurvey on IoT security: Application areas, security threats, and solution architectures," IEEE Access, vol. 7, no. 1, pp. 82721–82743, 2019.

[2] N. N. Tran, H. R. Pota, Q. N. Tran, and J. Hu, "Designing constraintbased false data injection attacks against the unbalanced distribution smart grids," IEEE Internet Things J., vol. 8, no. 11, pp. 9422–9435, Jun. 2021.

[3] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and openchallenges," Future Gener. Comput. Syst., vol. 82,pp. 395–411, May 2018. YIN et al.: NOVEL LENGTH-FLEXIBLE LIGHTWEIGHT CANCELABLE FINGERPRINT TEMPLATE 891

[4] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on Internet-scale IoT exploitations," IEEE Commun. Surveys Tuts., vol. 21, no. 3, pp. 2702–2733, 3rd Quart., 2019.

[5] K. Riad, R. Hamza, and H. Yan, "Sensitive and energetic IoT access control for managing cloud electronic health records," IEEE Access, vol. 7, pp. 86384–86393, 2019.

[6] O. J. A. Pinno, A. R. A. Grégio, and L. C. De Bona, "ControlChain: A new stage on the IoT access control authorization," Concurrency Comput. Pract. Exp., vol. 32, no. 12, p. e5238, 2020.

[7] A. K. Jain, K. Nandakumar, and A. Ross, "50 years of biometric research: Accomplishments, challenges, and opportunities," Pattern Recognit. Lett., vol. 79, pp. 80–105, Aug. 2016.

[8] X. F. Yin, Y. M. Zhu, and J. K. Hu, "Contactless fingerprint recognition based on global minutia topology and loose genetic algorithm," IEEE Trans. Inf. Forensics Security, vol. 15, pp. 28–41, 2020.

[9] M. S. Obaidat, S. P. Rana, T. Maitra, D. Giri, and S. Dutta, "Biometric security and Internet of Things (IoT)," in Biometric-Based Physical and Cybersecurity Systems. Cham, Switzerland: Springer, 2019, pp. 477–509.

[10] C.-X. Ren, Y.-B. Gong, F. Hao, X.-Y. Cai, and Y.-X. Wu, "When biometrics meet IoT: A survey," in Proc. Int. Asia Conf. Ind. Eng. Manage. Innov., 2016, pp. 635–643.

[11] X. Jiang et al., "Enhancing IoT security via cancelable HD-sEMG-based biometric authentication password, encoded by gesture," IEEE Internet Things J., vol. 8, no. 22, pp.16535–16547, Nov. 2021.