Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

Multi-Level Authentication: Combining Face, Palm, and Liveness Detection for Improved Security

Raja Suganya PV[1], Sam Joshua S[2], Vigneshwar B[3], Jai Kishan M[4]

[1]Asst Prof of Artificial Intelligence and Data Science, Velammal Engineering College, Chennai, Tamilnadu, India.

[2,3,4]Artificial Intelligence and Data Science, Velammal Engineering College, Chennai, Tamilnadu, India.

E-mail: [1]rajasuganya@velammal.edu.in, [2]samjoshua.s2002@gmail.com, [3]vigneshbw2002@gmail.com, [4]jaikishanmohan@gmail.com

Abstract

Face and palm recognition technologies have emerged as powerful tools for authentication, but they can still be vulnerable to fraud and impersonation. Liveness detection is a technique that can detect and prevent fraudulent attempts to bypass authentication by verifying the presence of a live human being during the authentication process. Combining face and palm recognition with liveness detection provides a highly effective and secure approach to authentication, which can prevent fraud and unauthorised access while providing a seamless and user-friendly experience.

Keywords: Face and palm recognition technologies, Liveness detection

1. Introduction

In today's digital age, security and privacy are paramount concerns for businesses and individuals alike. Authentication is one of the most common ways to ensure that only authorised individuals can access sensitive information or perform certain actions. However, traditional authentication methods like passwords or PINs can be vulnerable to fraud and theft and can be easily compromised.

DOI: https://doi.org/10.36548/jiip.2023.2.008
Received: 29.04.2023, received in revised form: 29.05.2023, accepted: 12.06.2023, published: 22.06.2023

To address these security concerns, face and palm recognition technologies have emerged as powerful tools for authentication. Facial and palm recognition technologies have become increasingly popular due to their convenience and accuracy. By using advanced algorithms and machine learning techniques, these technologies can accurately identify individuals based on their unique facial and palm features. However, even with these advanced technologies, there is still a risk of fraud or impersonation [1–4], which can compromise the security of the authentication process.

To address this issue, recent studies have demonstrated that the combination of facial and palm recognition with liveness detection can significantly enhance the security of the authentication process. This approach has shown great potential for preventing unauthorised access and fraudulent attempts. Despite its benefits, the integration of liveness detection into facial and palm recognition technologies poses some challenges. For example, it may require additional hardware or software components, which can increase the cost and complexity of the authentication system. Additionally, liveness detection techniques may raise privacy concerns as they involve the collection of additional data of the user.

2. Methodology

Liveness detection is a technique that verifies the presence of a live human being during the authentication process. It can detect and prevent fraudulent attempts to bypass authentication by using fake representations, such as photos or videos. By combining face and palm recognition with liveness detection, authentication systems can provide an added layer of security and ensure that only a living individual is granted access. Thus, the combination of face and palm recognition with liveness detection represents a highly effective and secure approach to authentication. This technology can prevent fraud and unauthorised access, while providing a seamless and user-friendly experience [5-6].

The flow chart of the proposed solution has been depicted in the below figure.

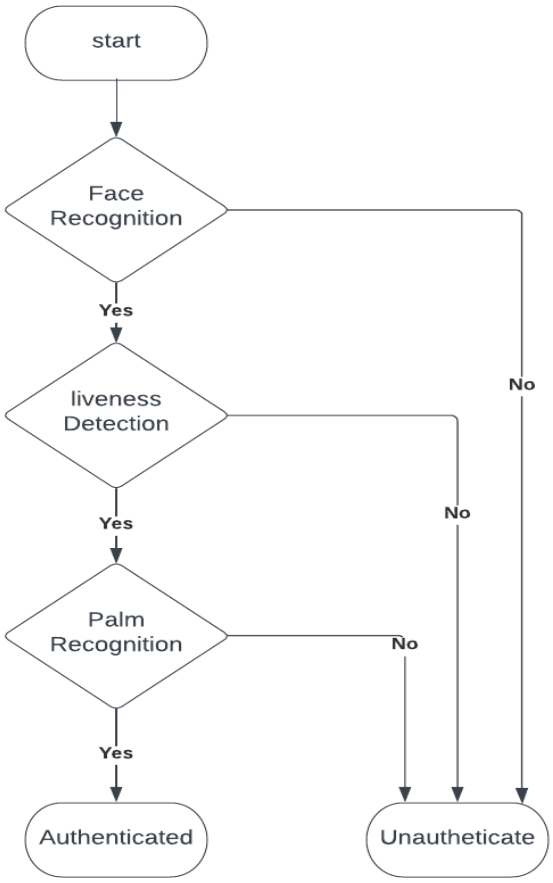

Figure 1. Flow chart of the Proposed Solution

A. Face Recognition

Face recognition has become an important research area in computer vision, with a wide range of applications in various fields. In recent years, there has been significant advancements in the face recognition technology, with the development of libraries and algorithms that can accurately identify individuals based on their facial features.

One such library is the Face Recognition Library, which provides a set of pre-trained models and functions for face recognition tasks. The library utilizes deep learning algorithms, specifically Convolutional Neural Networks (CNNs), to analyse and compare facial features. These models can be trained on large datasets, allowing them to identify a wide range of facial features with high accuracy.

The face recognition library also includes a set of pre-processing functions, that can be used to enhance facial images and improve the accuracy of recognition. These functions include techniques such as image alignment, normalisation, and feature extraction.

One of the most important applications of the face recognition library is in security and surveillance systems. With the ability to accurately identify individuals, these systems can prevent unauthorised access and detect suspicious behaviour. The library can also be used in other applications, such as facial recognition for social media or e-commerce platforms.

The steps involved in facial recognition are:

1. Set a time limit of 20 seconds for the function to run.
2. Initialize a video capture object using the specified camera.
3. Loop until the time limit is reached:
    a. Read a frame from the video capture object.
    b. Convert the frame to RGB format.
    c. Get the face encoding of the first face detected in the frame using the face_recognition library.
    d. Return the face encoding if a face is detected.
4. Return an empty list if no face is detected within the time limit.

B. Liveness Detection

Importing Required Libraries: The code starts by importing the required libraries. It imports the following libraries:

- cv2: OpenCV (Open-Source Computer Vision) library for real-time computer vision tasks, such as video capture and image processing.
- albumentations: A popular library for image augmentation that can be used in machine learning tasks.
- torch: PyTorch library for building and training deep learning models.

- datasouls_antispoof: A custom library that includes pre-trained models and class mappings.
- time: A built-in library in Python for measuring time.
- numpy: A library for numerical computations in Python.

Loading the pre-trained model: The code loads a pre-trained model called "tf_efficientnet_b3_ns" from the datasouls_antispoof library using the create_model function.

EfficientNet models are a family of neural network architectures designed for efficient use of computational resources while maintaining high performance on computer vision tasks. One variant, tf_efficientnet_b3_ns, is optimized for image classification tasks.

Liveness detection is the task of distinguishing between live and fake images or videos. To perform liveness detection using tf_efficientnet_b3_ns, a binary classifier can be trained on a dataset of both live and fake images and videos. Techniques such as printing photos or displaying videos on a screen and then recording them with a camera can be used to generate fake images and videos.

Transfer learning can then be used to fine-tune tf_efficientnet_b3_ns for liveness detection. The pre-trained weights of the model can be used as a starting point, and the model can be fine-tuned on the training dataset. During fine-tuning, the initial layers of the model can be frozen, and only later layers, specialised for image classification, can be trained. Additional layers can also be added to the model for further improvement.

The performance of the liveness detection model can be evaluated on a separate test dataset, and it can then be deployed in an application as needed.

Setting the Model to Evaluation Mode: The model.eval() method sets the model to the evaluation mode. This means that the model is not training, and that its parameters are fixed.

Defining the Liveness Function: The liveness function takes one argument, which is the camera number. It uses the cv2.VideoCapture method is used to capture a video from the specified camera. Then, it sets a time limit of 20 seconds using the t_end variable.

Reading Frames and Predicting: The function uses a while loop to read frames from the camera using the vid.read() method. Inside the loop, it converts the frame from BGR to RGB color space using the cv2.cvtColor method. It then applies a series of image augmentations using the albu.Compose method. These augmentations include padding the image to a minimum size of 400x400, center cropping the image to a size of 400x400, normalizing the pixel values, and converting the image to a PyTorch tensor using the ToTensorV2 method.

The model then makes a prediction on the preprocessed image using the torch.no_grad() method. The prediction is converted to a numpy array using the NumPy() method, and the argmax method is used to obtain the class with the highest probability.

If the predicted class is 0, the function returns true, indicating that the face is live. Otherwise, the loop continues until the time limit is reached. If no live face is detected within the time limit, the function returns false.

C. Hand Recognition

Palmprint recognition is a biometric technology that has received considerable attention in recent years. It has many applications in the fields of security and access control, as well as in forensic investigations. The use of deep learning techniques, specifically the Siamese network, has been proven to be effective in enhancing the accuracy and efficiency of palmprint recognition systems.

After identifying the bounding box around the hand, the code crops the image to only include the area within the box. The cropped image is then resized to a uniform size of 200x200 pixels, and the brightness levels are normalized using the histogram equalization technique. The resulting image is then saved to the disk as a JPEG file. The code appears to be part of a larger image processing pipeline, which may involve additional steps such as feature extraction, pattern recognition, or machine learning. Such pipelines can be used for a variety of applications.

Siamese networks are neural networks that use two or more identical subnetworks to learn the similarity or dissimilarity between two input samples. In palmprint recognition, a Siamese network can be used to extract feature representations from the palmprint images and then compare them with the representations of the images in the database. This approach helps

in detecting subtle differences between palmprints and makes the recognition process more accurate.

The use of the Siamese network in palmprint recognition has many advantages, including the ability to handle non-linear variations in palmprint images, and the ability to work with a small amount of training data. It also enables the development of robust and scalable palmprint recognition systems that can be used in a wide range of applications.

In conclusion, palmprint recognition using the Siamese network is a promising technology that has the potential to enhance the accuracy and efficiency of palmprint recognition systems. The use of deep learning techniques such as Siamese network can lead to the development of more robust and scalable systems, that can be used in a wide range of applications.

Figure 2. Palm Authentication Results

3. Results

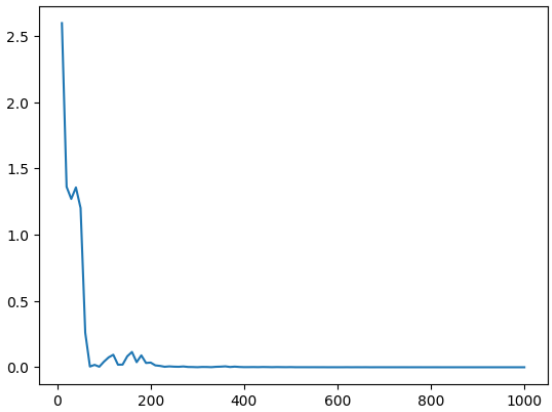

Figure 3. Palm Authentication Epoch Vs Error

This finding suggests that the proposed approach achieves superior accuracy in comparison to prior methods that may not have utilized extended training epochs.

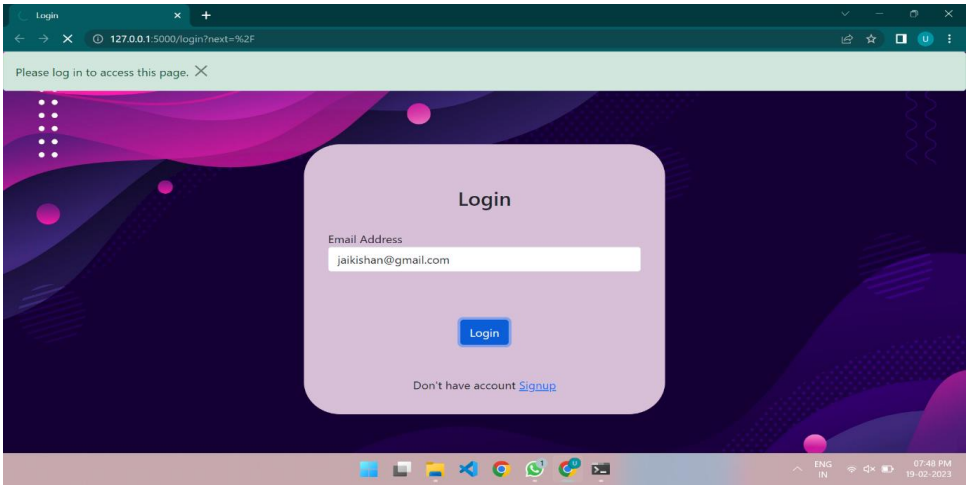

Figure 4. Login Page

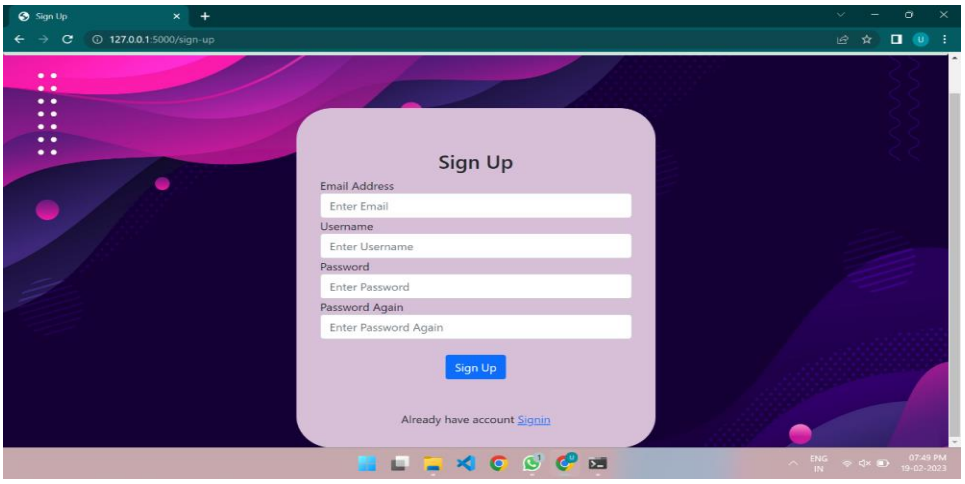

Figure 5. Sign UP page

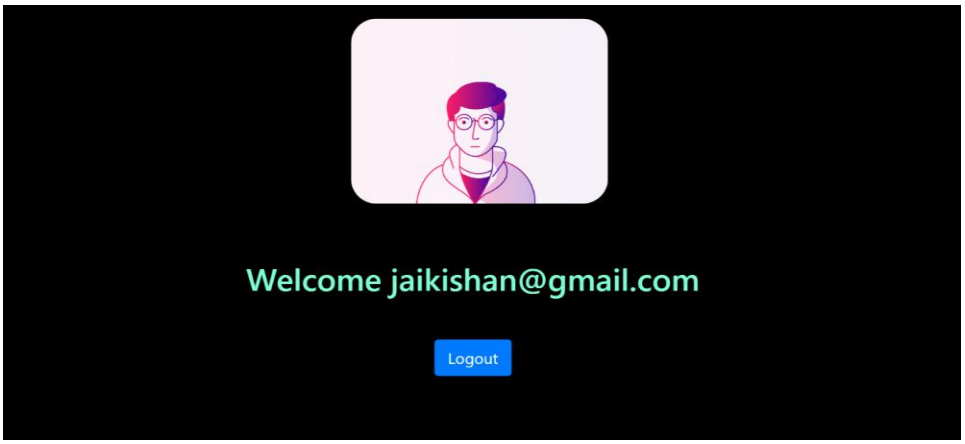

Figure 6. Index Page

4. Conclusion

As the world becomes more digitalized, the importance of securing sensitive information and data has become increasingly important. Combining face and palm recognition with liveness detection provides a highly effective and secure approach to authentication, which can prevent fraud and unauthorized access, while providing a seamless and user-friendly experience. The combination of these technologies has emerged as a powerful tool to enhance security and privacy in both personal and business settings.

References

[1] A. Geitgey, "Face recognition library," Available online: https://github.com/ageitgey/face_recognition

[2] Y. Taigman, et al., "Deep Face Recognition: A Survey," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1701-1708, doi: 10.1109/CVPR.2014.220.

[3] F. Schroff, et al., "FaceNet: A Unified Embedding for Face Recognition and Clustering," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 815-823, doi: 10.1109/CVPR.2015.7298682.

[4] K. He, et al., "Deep Residual Learning for Image Recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[5] A. Rathgeb, F. Breitinger, and C. Busch, "Liveness Detection in Biometrics," in IEEE Transactions on Information Forensics and Security, vol. 10, no. 4, pp. 844-861, April 2015, doi: 10.1109/TIFS.2015.2414623.

[6] Q. Zhao, et al., "Real-Time Face Recognition with Deep FaceNet," in 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN), 2017, pp. 682-686, doi: 10.1109/ICCSN.2017.8230254.

[7] A. Das, et al., "Advances in Face Recognition: A Comprehensive Review," in IET Biometrics, vol. 8, no. 1, pp. 9-50, Jan. 2019, doi: 10.1049/iet-bmt.2017.0225.

[8] X. Tan, et al., "Face Detection and Recognition Using Deep Learning: A Survey," in International Journal of Pattern Recognition and Artificial Intelligence, vol. 33, no. 01, 2019, 1942002, doi: 10.1142/S0218001419420021.

[9] Y. Taigman, et al., "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1701-1708, doi: 10.1109/CVPR.2014.220.

[10] Z. Liu, et al., "Large-scale CelebFaces Attributes (CelebA) Dataset," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 5045-5053, doi: 10.1109/CVPR.2015.7299102.

[11] R. Rothe, R. Timofte, and L. Van Gool, "Age and Gender Estimation of Unfiltered Faces," in IEEE International Conference on Computer Vision (ICCV), 2015, pp. 387-391, doi: 10.1109/ICCV.2015.56.

[12] Y. Taigman, et al., "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1701-1708, doi: 10.1109/CVPR.2014.220.

[13] Amos, et al., "OpenFace: A General-Purpose Face Recognition Library with Mobile Applications," in CMU-CS-16-118, CMU School of Computer Science, 2016, Available online: http://cmusatyalab.github.io/openface/

[14] X. Cao, Y. Wei, F. Wen, and J. Sun, "Face Alignment by Explicit Shape Regression," in International Journal of Computer Vision, vol. 107, no. 2, pp. 177-190, 2014, doi: 10.1007/s11263-013-0676-x.

[15] X. Wu, et al., "LightCNN: Deeply Learning Face Representations," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1288-1296, doi: 10.1109/ICCV.2015.156.

[16] Jain, A.K., Ross, A. and Nandakumar, K. (2016) 'Palmprint recognition and its applications', Springer International Publishing, Switzerland, doi: 10.1007/978-3-319-29854-2_2.

Author's Biography

Raja Suganya PV[1] - Assistant Professor of Artificial Intelligence and Data Science at Velammal Engineering College, Chennai. She has a year of experience in teaching and has produced above 95% results. She has played a vital role in the department duties such as Exam cell in charge, counseling, and mentoring of students' academic progress, as well as organizing workshops and FDP. Her area of interest is Data Mining, and she has conducted research in this field.

Sam Joshua S[2] is a graduate student pursuing a degree in Artificial Intelligence and Data Science at Velammal Engineering College, Chennai. He has a strong passion for research in the field of machine learning and data science. His current research interests include deep learning and natural language processing.

Vigneshwar B[3] is an undergraduate student majoring in Artificial Intelligence and Data Science at Velammal Engineering College, Chennai. He is a motivated and enthusiastic individual with a keen interest in exploring the applications of AI and data science in various domains. His research interests include computer vision and big data analytics.

Jai Kishan M[4]is an undergraduate student pursuing a degree in Artificial Intelligence and Data Science at Velammal Engineering College, Chennai. He has a strong foundation in programming and data structures and is interested in applying his skills to real-world problems. His research interests include natural language processing and reinforcement learning.