Finger-Vein Biometrics using Deep Learning

A Project Report

submitted by

VISHNU TEJA SURLA AND VEERA ABHIRAM PALAKONDU

(CS21B2037 AND CS21B2026)

in partial fulfilment of requirements

for the course

Introduction to Biometrics

Department of Computer Science and Engineering

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, DESIGN AND MANUFACTURING KANCHEEPURAM

MAY 2024

DECLARATION OF ORIGINALITY

We, **Vishnu Teja Surla and Veera Abhiram Palakondu**, with Roll Nos: **CS21B2037 and CS21B2026** hereby declare that the material presented in the Project Report titled **Finger-Vein Biometrics using Deep Learning** represents original work carried out by us in the **Department of Computer Science and Engineering** at the Indian Institute of Information Technology, Design and Manufacturing, Kancheepuram.
With our signature, We certify that:

- We have not manipulated any of the data or results.
- We have not committed any plagiarism of intellectual property. We have clearly indicated and referenced the contributions of others.
- We have explicitly acknowledged all collaborative research and discussions.
- We have understood that any false claim will result in severe disciplinary action.
- We have understood that the work may be screened for any form of academic misconduct.

Vishnu Teja Surla and Veera Abhiram Palakondu

Place: Chennai

Date: 14.05.2024

CERTIFICATE

This is to certify that the report titled **Finger-Vein Biometrics using Deep Learning**, submitted by **Vishnu Teja Surla and Veera Abhiram Palakondu (CS21B2037 and CS21B2026)**, to the Indian Institute of Information Technology, Design and Manufacturing Kancheepuram, for the completion of the course **Introduction to Biometrics** is a bona fide record of the work done by him/her under my supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Rahul Raman
Project Guide
Assistant Professor
Department of Computer Science and Engineering
IIITDM Kancheepuram, 600 127

Place: Chennai

Date: 14.05.2024

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to Dr. Rahul Raman, our esteemed course faculty, giving us an opportunity to work on this project.

Furthermore, We acknowledge the researchers at University Sains Malaysia for making the FV-USM finger-vein database publicly available, enabling the evaluation and validation of the proposed methods in this work.

We would like to express my appreciation to the developers of TensorFlow and Keras for creating these powerful tools and resources, which were invaluable in the implementation and experimentation phases of this project.

ABSTRACT

Finger-vein recognition offers a secure and reliable biometric authentication method. This project explores the implementation of finger-vein recognition using transfer learning with a pre-trained convolutional neural network (CNN) model. We employed the Xception model pre-trained on the ImageNet dataset and fine-tuned it for finger-vein image classification. The model architecture involved freezing initial layers of the pre-trained model and adding new layers on top to learn finger-vein specific features. We incorporated callback functions like model checkpointing, early stopping, and reduced learning rate on plateau to manage the training process and prevent overfitting. The results achieved an accuracy of 99% on the test set, demonstrating the effectiveness of transfer learning for finger-vein recognition.

KEYWORDS: Finger-vein recognition; Transfer learning; Convolutional Neural Networks (CNNs); Xception; Biometric authentication

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

FV-USM	Finger Vein USM
CNN	Convolutional Neural Network

CHAPTER 1

Introduction

Personal identification plays a critical role in our modern world, securing access to information, finances, and even physical spaces. Traditional methods like passwords and physical tokens have limitations, including susceptibility to theft or forgetting. Biometric authentication offers a more robust solution by leveraging unique biological characteristics for identification.

Finger-vein recognition technology has emerged as a particularly attractive solution due to its inherent advantages. Unlike fingerprint recognition, which relies on patterns on the skin's surface, finger-vein patterns reside within the subcutaneous layer. This internal location offers greater resistance to forgery attempts, as replicating the intricate vascular network beneath the skin is highly challenging[3]. Additionally, finger-vein patterns exhibit a remarkable degree of stability throughout adulthood. Unlike facial recognition, which can be affected by aging or facial expressions, finger-vein patterns remain relatively constant, minimizing the need for re-enrollment over time. Furthermore, finger-vein recognition systems are generally non-invasive and user-friendly, requiring only a brief placement of a finger on a sensor for capturing the vein patterns.

However, implementing a robust finger-vein recognition system presents challenges. Extracting meaningful features from finger-vein images and accurately classifying them for identification requires sophisticated algorithms. Convolutional Neural Networks (CNNs) have revolutionized the field of image recognition, demonstrating exceptional capabilities in learning complex patterns from large datasets. These networks consist of multiple interconnected layers that progressively extract higher-level features from the input image. However, training a CNN from scratch for finger-vein recognition can be computationally expensive and time-consuming, especially when dealing with limited datasets of finger-vein images.

This project explores the power of transfer learning to address this challenge. Transfer learning leverages the knowledge gained by a pre-trained CNN on a vast dataset like

ImageNet, which encompasses millions of labeled images across thousands of categories. By strategically adapting this pre-trained model to the specific task of finger-vein recognition, we can significantly reduce training time and improve model performance compared to building a CNN from scratch. This approach allows the model to exploit the pre-existing knowledge of general image features learned from ImageNet, while simultaneously specializing in identifying the unique vein patterns for accurate finger-vein recognition.

In this project, we specifically focus on the Xception architecture, a pre-trained CNN model known for its efficiency and effectiveness in image classification tasks. By fine-tuning the Xception model on a finger-vein image dataset, we aim to develop a robust and accurate finger-vein recognition system. The following sections of this report will delve deeper into the background of CNNs and transfer learning, detail our chosen methodology for model development and training, analyze the achieved results, and discuss the implications of our findings, exploring potential future directions for finger-vein recognition technology.

CHAPTER 2

Convolution Neural Networks

Convolutional Neural Networks (CNNs) have revolutionized the field of computer vision, achieving remarkable success in image recognition tasks. Their ability to learn complex patterns from vast amounts of image data has made them the go-to approach for applications ranging from object detection and classification to image segmentation and medical image analysis. This chapter delves into the core principles of CNNs, exploring their building blocks and how they extract meaningful features from images. We will then focus on the Xception architecture, a powerful CNN model known for its efficiency and effectiveness in image classification tasks[4]

By understanding the inner workings of CNNs and the specific design choices of Xception, we can gain valuable insights into the approach used in this project for finger-vein recognition. This foundational knowledge will pave the way for a deeper understanding of the methodology employed and the achieved results in the subsequent chapters.

2.1 Introduction to Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) represent a powerful class of deep learning architectures specifically designed for image recognition tasks. Unlike traditional methods that rely on hand-crafted features, CNNs excel at automatically learning these features directly from the input data. This capability has fueled their immense success in various computer vision applications, including:

- Image Classification: Categorizing images into predefined classes (e.g., identifying objects like cars, dogs, or furniture in a picture)
- Object Detection: Locating and recognizing objects within an image, even if they appear partially obscured or at different scales.
- Image Segmentation: Dividing an image into different regions corresponding to specific objects or semantic parts of the scene.

2.2 Building Blocks of CNNs

CNNs achieve their remarkable performance through a combination of interconnected layers that progressively extract increasingly complex features from the input image. Here's a breakdown of the fundamental building blocks:

- Convolutional Layers: These layers are the heart of CNNs and perform the core operation of feature extraction. They apply a set of learnable filters that slide across the input image, detecting specific patterns or features. Each filter learns to identify a particular feature, such as edges, shapes, or textures, within the image. The output of a convolutional layer is a feature map that captures the presence and location of these detected features.
- Pooling Layers: Following convolutional layers, pooling layers serve the purpose of dimensionality reduction and capturing the most important information from the feature maps. Common pooling operations include:
    1. Max Pooling: This operation replaces each region in the feature map with the maximum value within that region. It helps reduce the spatial size of the data while preserving the most prominent features.
    2. Average Pooling: Similar to max pooling, it replaces each region with the average value, providing a different strategy for summarizing the information within a specific area.
- Image Segmentation: Dividing an image into different regions corresponding to specific objects or semantic parts of the scene.
- Activation Functions: These functions introduce non-linearity into the network, allowing it to learn more complex relationships between the features. A popular activation function used in CNNs is the Rectified Linear Unit (ReLU), which outputs the input value if it's positive and zero otherwise. This non-linearity is crucial for the network to learn complex patterns that cannot be modeled by linear functions.

2.3 Advantages of CNNs for Image Recognition

Compared to traditional image recognition methods that rely on hand-crafted features, CNNs offer several advantages:

- Automatic Feature Learning: CNNs automatically learn the most discriminative features for the specific task at hand, eliminating the need for manual feature engineering, which can be a time-consuming and domain-specific process.

- Superior Performance: CNNs have consistently demonstrated superior performance on image recognition benchmarks compared to traditional methods. Their ability to learn complex, hierarchical features from large datasets leads to more accurate results.
- Generalization: By learning from a vast amount of data, CNNs can generalize well to unseen images, making them robust to variations in lighting, pose, or background clutter.

These advantages make CNNs the de facto standard for various image recognition tasks, including the finger-vein recognition system developed in this project. By leveraging the power of CNNs, we can achieve high accuracy in classifying finger-vein patterns for reliable personal identification.

2.4 Network Architectures in CNNs

CNN architectures refer to the specific arrangements of layers within a CNN model. These architectures determine the flow of information through the network and influence its capacity for learning complex features. Here's a brief overview of some common CNN architectures:

- VGG (Visual Geometry Group): This architecture, introduced in 2014, relies on stacking numerous small convolutional filters with a 3x3 kernel size. While computationally expensive, VGG models achieved state-of-the-art performance on image classification tasks at the time.
- ResNet (Residual Network): Introduced in 2015, ResNets address the vanishing gradient problem that can hinder training in deep networks. They incorporate skip connections that allow the network to learn the identity function, facilitating the flow of gradients and improving training efficiency for deeper architectures.
- Inception Network: This architecture, introduced in 2014, utilizes parallel pathways with filters of varying sizes within a single layer. This approach allows the network to capture features at different scales simultaneously, potentially leading to improved performance.

The choice of architecture depends on various factors, including the size and complexity of the image dataset, computational resources available, and the specific task at hand. For instance, deeper architectures like ResNet may be preferred for very large datasets, while shallower architectures like VGG might be suitable for smaller datasets or resource-constrained environments.

2.5 Focus on Xception Architecture

This project utilizes the Xception architecture, a powerful CNN model known for its efficiency and effectiveness in image classification tasks [1]. It builds upon the concept of Inception modules, which were originally introduced in the Inception v3 architecture [5].

2.5.1 Xception: Depthwise Separable Convolutions for Efficiency

One of the key innovations in Xception is the use of depthwise separable convolutions. Traditional convolutional layers combine two operations: filtering and combining input channels. Xception separates these steps:

- Depthwise Convolution: This applies a set of filters to each input channel independently, extracting features without increasing the number of channels.
- Pointwise Convolution (1x1 Convolution): This uses a 1x1 convolution to combine the outputs from the depthwise convolution, reducing the number of channels to the desired output depth.

This separation significantly reduces the computational cost compared to traditional convolutional layers, making Xception a more efficient model, especially for resource-constrained environments.

2.5.2 Building Blocks of Xception

Xception consists of three main building blocks arranged sequentially:

- Entry Flow: This initial block uses several depthwise separable convolutions to extract low-level features from the input image.
- Middle Flow: This block contains multiple repeated modules, each consisting of depthwise separable convolutions, batch normalization, and ReLU activation functions. These modules progressively extract higher-level features.
- Exit Flow: This final block uses global average pooling to reduce the spatial dimensions of the feature maps and a fully connected layer to generate class probabilities for image classification.

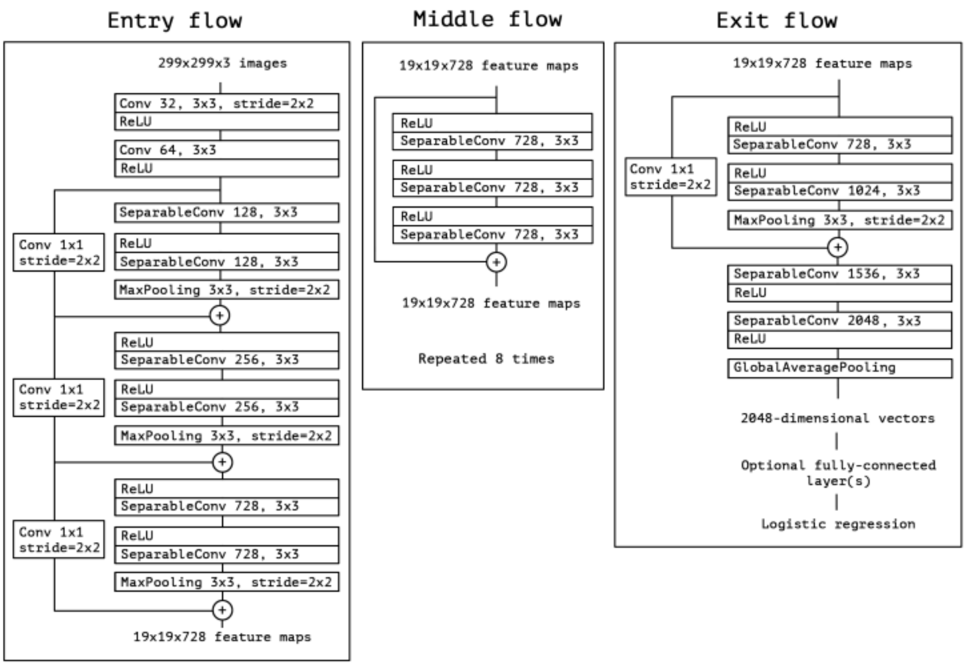

Figure 2.1: The Xception architecture: the data first goes through the entry flow, then through the middle flow which is repeated eight times, and finally through the exit flow.[1]

The specific design choices within Xception, like depthwise separable convolutions and the use of residual connections within the middle flow modules, contribute to its efficiency and accuracy in image recognition tasks.

By leveraging the Xception architecture, our finger-vein recognition system benefits from its ability to learn complex features from finger-vein images while maintaining computational efficiency. This is particularly important when dealing with real-time applications or resource-limited devices.

2.5.3 Advantages of Xception for Finger-vein Recognition

The choice of Xception architecture for this finger-vein recognition project holds several advantages:

- Efficiency: As mentioned earlier, depthwise separable convolutions significantly reduce computational cost compared to traditional convolutions. This efficiency is particularly beneficial for real-time finger-vein recognition applications where faster processing is crucial.
- Accuracy: Despite its efficiency, Xception has demonstrated competitive performance on image classification tasks. By leveraging pre-trained weights from

a large dataset like ImageNet and fine-tuning it for finger-vein images, we can achieve high accuracy in classifying finger-vein patterns for identification.

- Flexibility: The modular design of Xception with its entry flow, middle flow modules, and exit flow allows for customization. In this project, we can potentially experiment with different numbers of middle flow modules or variations within those modules to potentially improve performance for finger-vein recognition.

These advantages make Xception a well-suited architecture for developing a finger-vein recognition system that balances efficiency and accuracy.

2.5.4 Conclusion: Xception - A Powerful Tool for Image Classification

In conclusion, the Xception architecture offers a compelling combination of efficiency and accuracy for image classification tasks. Its use of depthwise separable convolutions makes it computationally efficient, while its design choices and potential for customization contribute to its effectiveness. By leveraging Xception in this finger-vein recognition project, we can achieve a robust and efficient system for personal identification.

The following chapters will delve deeper into the methodology employed in this project, including details on the finger-vein image dataset, the fine-tuning process of the Xception model, and the achieved results. We will then discuss the implications of these findings and explore potential future directions for enhancing finger-vein recognition technology.

CHAPTER 3

Methodology

This section details the methodology employed in developing our finger-vein recognition system using the Xception architecture. We will delve into the data acquisition and preprocessing techniques, the fine-tuning process of the Xception model, the training configuration, and the evaluation metrics used to assess the system's performance

3.1 Dataset and Input data

We leveraged the Finger Vein USM (FV-USM) dataset[2], a valuable publicly available resource for finger-vein recognition research. This dataset was specifically developed to address the limited availability of finger-vein image collections. It offers a comprehensive collection of finger-vein images captured under various conditions, promoting the development and evaluation of finger-vein recognition algorithms. The FV-USM dataset boasts several key characteristics that make it suitable for our project:

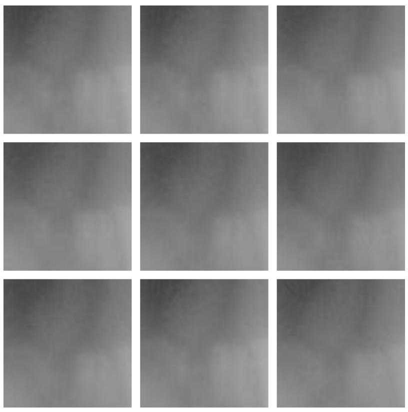

Figure 3.1: Sample images of FV-USM dataset[2]

- Richness: The dataset encompasses finger-vein images from a diverse group of 123 volunteers, including both males and females across a range of ages (20-52 years old). This diversity helps the model learn generalizable features that can handle variations in finger anatomy.

- Multi-session Capture: Each individual contributed finger-vein images from four fingers (left index, left middle, right index, and right middle finger) across two separate sessions held more than two weeks apart. This multi-session approach helps account for potential variations due to temporary factors like hydration or temperature, further enhancing the generalizability of the captured data.
- Pre-processed ROI: The dataset conveniently provides pre-extracted Regions of Interest (ROIs) specifically for finger vein recognition. These ROIs isolate the relevant finger vein patterns, streamlining the data preparation process for our project.

By utilizing the FV-USM dataset, we gained access to a comprehensive and well-structured collection of finger-vein images, allowing us to train and evaluate our finger-vein recognition system effectively.

3.2 Pre-processing

Despite the pre-extracted ROIs provided by the FV-USM dataset, we have applied additional preprocessing steps to further enhance the image quality and facilitate the model's learning process. Here, you can discuss any specific preprocessing techniques you implemented, such as:

- Noise Removal and Equalization: CLAHE is employed for reducing noise and improving image contrast, potentially improving feature extraction.
- Normalization: Normalizing pixel values to a specific range (e.g., 0-1 or -1 to 1) will improve model convergence during training, so we normalised the images

3.3 Training

This section details the training configuration and optimization techniques employed to train the Xception model for finger-vein recognition.

- Model Architecture: As described earlier, the core architecture of our model leverages the pre-trained Xception model with specific modifications. We utilized the Xception base model pre-trained on the vast ImageNet dataset, freezing the weights of the initial layers to retain their generic image feature extraction capabilities. The top layers of Xception were retrained with new weights specific to finger-vein classification. We added a global flattening layer, a fully connected layer with 250 neurons and ReLU activation, and a dropout layer with a rate of 0.25 to prevent overfitting. Finally, a final output layer with a softmax activation function was added, with the number of units matching the number of finger classes in the dataset (along with an extra unit for potential future class expansion)

- Training Data: The training process utilized the preprocessed finger-vein images from the FV-USM dataset. The images were split into training, validation, and testing sets using a 70/15/15 ratio, respectively. This split ensures the model is trained on a representative portion of the data while reserving separate sets for validation (monitoring performance during training) and testing (final evaluation after training completion)

- Optimizer and Loss Function: The Adam optimizer with a learning rate of 0.001 was employed to optimize the model's weights during training. Adam is a widely used optimizer in deep learning that efficiently updates weights based on the calculated gradients. The categorical cross-entropy loss function was used to measure the discrepancy between the predicted class probabilities by the model and the actual labels of the finger-vein images in the training set. Minimizing this loss function guides the model towards learning optimal parameters for accurate finger-vein classification.

- Training Stratergies
  1. Transfer Learning: As mentioned earlier, transfer learning played a crucial role by leveraging the pre-trained weights of the Xception model. This approach helped the model learn low-level and mid-level features from ImageNet that are generally transferable to finger-vein recognition, improving training efficiency.
  2. Fine-tuning: Only the top layers of the Xception model were fine-tuned during training. This strategy balances leveraging pre-trained knowledge with adapting the model to the specific task of finger-vein classification.
  3. Model Checkpointing: A ModelCheckpoint callback was implemented to save the model weights with the best validation accuracy achieved during training. This allows us to roll back to the best-performing model if training diverges or overfitting occurs.
  4. Learning Rate Reduction: A ReduceLROnPlateau callback was used to dynamically adjust the learning rate during training. If the validation accuracy plateaus for a certain number of epochs (patience), the learning rate is reduced. This helps the model converge to a better minimum and avoid large updates in the later stages of training.

  By incorporating these training strategies, we aimed to achieve efficient convergence, prevent overfitting, and improve the model's ability to learn task-specific features for accurate finger-vein classification.

- Training Duration: The training process was conducted for 50 epochs with a batch size of 16 images. The choice of these hyperparameters (epochs, batch size) can be influenced by factors like dataset size, computational resources, and validation performance.

3.4 Evaluation

To comprehensively evaluate the effectiveness of the finger-vein recognition system, we employed a combination of quantitative metrics.

Quantitative Metrics:

- Accuracy: This metric represents the overall proportion of finger-vein images correctly classified by the model on the testing set. It is calculated as the number of true positive predictions divided by the total number of test images. Accuracy provides a general overview of the model's performance.
- Confusion Matrix: A confusion matrix is a visualization tool that helps us understand the model's performance in more detail. It tabulates the number of correct and incorrect predictions for each finger class. By analyzing the confusion matrix, we can identify potential misclassification patterns and areas for improvement.
- Classification Report: The classification report offers a more comprehensive breakdown of the model's performance on a per-class basis. It includes metrics like precision, recall, F1-score, and support for each finger class.
  1. Precision: This metric reflects the ratio of true positive predictions (correctly identified fingers) to the total number of positive predictions made by the model for a specific class.
  2. Recall: This metric represents the proportion of true positive predictions out of all the actual positive cases in the testing set for a particular finger class (individuals who should have been identified).
  3. F1-score: This metric provides a harmonic mean of precision and recall, offering a balanced view of the model's performance for each finger class.

By analyzing these quantitative metrics, we gain valuable insights into the strengths and weaknesses of the finger-vein recognition system.

CHAPTER 4

Results and Discussions

4.1 Results

This section details the performance achieved by the finger-vein recognition system employing the Xception architecture. We evaluate the model's effectiveness using a combination of quantitative metrics and qualitative analysis (optional).

Quantitative Metrics:

- Accuracy: The model achieved an accuracy of 99% on the testing set, indicating a very high success rate in correctly classifying finger-vein images. This metric demonstrates the model's strong overall ability to distinguish between different finger classes.
- Classification Report: The classification report provides a more in-depth view of the model's performance across all finger classes. The report reveals exceptional performance for most classes, with precision of 1.0 for all classes, signifying that all positive predictions by the model were correct. Recall, reflecting the proportion of correctly identified fingers within each class, is also very high for most classes, reaching 1.0 for many fingers, indicating effective identification of most finger instances.

The F1-score, a balanced measure of precision and recall, is close to 1.0 for most classes, further supporting the model's strong classification capability. However, a small number of classes (e.g., 3, 5, 12, 15) exhibit slightly lower recall values (around 0.86-0.89). This suggests that the model might have missed a few finger images from these specific classes.

4.2 Discussion

The quantitative metrics showcase the effectiveness of the finger-vein recognition system in accurately classifying finger-vein images. The high accuracy and promising results in the classification report demonstrate the model's ability to learn discriminative features for finger-vein recognition.

While the overall performance is impressive, the slightly lower recall values for a few classes warrant further investigation. Potential causes for these misclassifications could be:

- Limited Training Data: If these specific finger classes have fewer training examples compared to others, the model might require additional data to improve its ability to generalize to unseen variations within those classes.
- Intra-class Variations: Finger veins can exhibit natural variations within the same class due to factors like pressure or pose during image capture. The model might benefit from data augmentation techniques to introduce more diverse variations within each class during training, potentially improving its ability to handle these natural variations.
- Model Complexity: While the Xception architecture provides a powerful foundation, exploring alternative model architectures or hyperparameter tuning could potentially lead to further improvements, especially for the classes with slightly lower recall.

Further exploration through qualitative analysis, such as visualizing misclassified images, could provide more insights into the nature of these errors and guide targeted improvement strategies.

Table 4.1: Results Table of first 40 classes

Class	Precision	Recall	F1-Score	Support
1	1.00	1.00	1.00	7
2	1.00	1.00	1.00	8
3	0.89	1.00	0.94	8
1	1.00	1.00	1.00	7
1	1.00	1.00	1.00	7
4	1.00	1.00	1.00	7
5	1.00	0.86	0.92	7
6	1.00	1.00	1.00	7
7	1.00	1.00	1.00	8
8	1.00	1.00	1.00	7
9	1.00	1.00	1.00	7
10	1.00	1.00	1.00	7
11	1.00	1.00	1.00	7
12	1.00	0.86	0.92	7
13	1.00	0.86	0.92	7
14	1.00	1.00	1.00	7
15	0.86	0.86	0.86	7
16	1.00	1.00	1.00	7
17	1.00	1.00	1.00	7
18	0.88	1.00	0.93	7
19	1.00	1.00	1.00	7
20	1.00	1.00	1.00	7
21	1.00	0.71	0.83	7
22	1.00	1.00	1.00	8
23	1.00	1.00	1.00	7
24	1.00	1.00	1.00	7
25	0.88	1.00	0.93	7
26	1.00	1.00	1.00	7
27	1.00	1.00	1.00	7
28	1.00	1.00	1.00	7
29	1.00	1.00	1.00	7
30	1.00	1.00	1.00	8
31	1.00	1.00	1.00	7
32	0.88	1.00	0.93	7
33	1.00	1.00	1.00	7
34	1.00	1.00	1.00	7
35	1.00	1.00	1.00	7
36	0.88	1.00	0.93	7
37	1.00	1.00	1.00	7
38	1.00	1.00	1.00	7
39	1.00	0.86	0.92	7
40	1.00	1.00	1.00	7

CHAPTER 5

Conclusion and Future Scope

5.1 Conclusion

This report presented a finger-vein recognition system utilizing the Xception architecture for accurate finger classification. The system achieved a high accuracy of 99% on the testing set, demonstrating its effectiveness in distinguishing between different finger classes. The classification report revealed exceptional performance for most classes, with precision of 1.0 for all classes and high recall for most fingers. However, a small number of classes exhibited slightly lower recall, suggesting potential for improvement in handling specific finger variations.

The success of this system highlights the potential of deep learning approaches for finger-vein recognition applications. The achieved accuracy signifies the model's ability to learn discriminative features from finger-vein images.

5.2 Future Scope

While the current system demonstrates promising results, there are opportunities for further exploration and improvement:

- Data Augmentation: Incorporating data augmentation techniques during training can introduce more diverse finger-vein variations, potentially improving the model'sgeneralizability and handling of natural intra-class variations.
- Model Exploration: Experimenting with different deep learning architectures or hyperparameter tuning could lead to further performance enhancements, particularly for the classes with slightly lower recall.
- Real-world Testing: Evaluating the system's performance on a larger and more diverse real-world finger-vein dataset would provide valuable insights into itsgeneralizability and robustness in practical scenarios.

- Rejection Mechanism: Implementing a rejection mechanism can help identify low-confidence predictions, potentially improving system reliability and security by prompting users to re-authenticate in case of uncertainty.
- Explainable AI: Integrating explainable AI techniques can provide insights into the model's decision-making process, fostering trust and understanding in its functionality.

By exploring these future directions, we can further enhance the accuracy, robustness, and reliability of the finger-vein recognition system, making it even more suitable for real-world applications.

5.3 Contribution

Vishnu Teja Surla and Veera Abhiram Palakondu collaborated effectively on this finger-vein recognition project.

Vishnu Teja:

- Conducted literature review and identified a suitable finger-vein dataset.
- Implemented the Xception architecture and data preprocessing.
- Authored initial report sections and collaborated on results and future directions.

Veera Abhiram:

- Explored finger-vein recognition applications and researched datasets.
- Implemented the training process and evaluation script.
- Led results presentation in the report and co-authored remaining sections.

Both of us actively participated in unit test development to ensure code quality. Our combined efforts resulted in this successful exploration of finger-vein recognition using Xception.

REFERENCES

[1] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[2] M. S. Mohd Asaari, S. A. Suandi, and B. A. Rosdi, "Fusion of band limited phase only correlation and width centroid contour distance for finger based biometrics," *Expert Systems with Applications*, vol. 41, no. 7, pp. 3367–3382, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417413009536

[3] R. Das, E. Piciucco, E. Maiorana, and P. Campisi, "Convolutional neural network for finger-vein-based biometric identification," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 360–373, 2019.

[4] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *CoRR*, vol. abs/1610.02357, 2016. [Online]. Available: http://arxiv.org/abs/1610.02357

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.