Learning Ridge Structures: Adaptive CNN-Based Local Orientation and Frequency Estimation for Fingerprints

A Project Report submitted in the partial fulfillment of the Requirements for the award of the degree

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Atmakuri Jashwanth (22471A05L3)
Shaik Inkollu Farzan Basha (22471A05P1)
Yalagala Leela krishna (22471A05P2)

Under the esteemed guidance of

Mrs. Syed Rizwana, M.Tech., (Ph.D).

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASAROPET

(AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA under

Tyre -1 and an ISO 9001:2015 Certified

Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada

KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET- 522601

2025-2026

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project that is entitled with the name "Learning Ridge Structures: Adaptive CNN-Based Local Orientation and Frequency Estimation For Fingerprints" is a bonafide work done by Atmakuri Jashwanth (22471A05L3), Shaik Inkollu Farzan Basha (22471A05P1), Yalagala Leela krishna (22471A05P2) in partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2025-2026.

PROJECT GUIDE
Mrs. Syed Rizwana, M.Tech., (Ph.D).
Assistant Professor

PROJECT CO-ORDINATOR
Mrs. Syed Rizwana, M.Tech., (Ph.D).
Assistant Professor

HEAD OFTHE DEPARTMENT
Dr. S. N. Tirumala Rao, M.Tech., Ph.D.
Professor & HOD

EXTERNAL EXAMINER

DECLARATION

We declare that this project work titled **"Learning Ridge Structures: Adaptive CNN-Based Local Orientation and Frequency Estimation for Fingerprints "** is composed by me that the work contain here is my own except where explicitly stated otherwise in the text and that this work has not been submitted for any other degree or professional qualification except as specified.

Atmakuri Jashwanth (22471A05L3)

Shaik Inkollu Farzan Basha (22471A05P1)

Yalagala Leela krishna (22471A05P2)

ACKNOWLEDGEMENT

We wish to express my thanks to various personalities who are responsible for the completion of my project. I am extremely thankful to my beloved chairman**, Sri M. V. Koteswara Rao, B.Sc.,** who took keen interest in me in every effort throughout this course. I owe my sincere gratitude to my beloved principal**, Dr. S. Venkateswarlu, Ph.D.,** for showing his kind attention and valuable guidance throughout the course.

We express our deep-felt gratitude towards **Dr. S. N. Tirumala Rao, M.Tech., Ph.D.,** HOD of the CSE department, and also to my guide**, Mrs. Syed Rizwana, M.Tech., (Ph.D).,** Assistant Professor of the CSE department, whose valuable guidance and unstinting encouragement enabled me to accomplish my project successfully in time.

We extend our sincere thanks to **Mrs. Syed Rizwana, M.Tech., (Ph.D.),** Assistant Professor & Project Coordinator of the project, for extending her encouragement. Their profound knowledge and willingness have been a constant source of inspiration for me throughout this project work.

We extend our sincere thanks to all the other teaching and non-teaching staff in the department for their cooperation and encouragement during my B.Tech. degree.

We have no words to acknowledge the warm affection, constant inspiration, and encouragement that I received from my parents.

We affectionately acknowledge the encouragement received from my friends and those who were involved in giving valuable suggestions and clarifying my doubts, which really helped me in successfully completing my project.

By

Atmakuri Jashwanth (22471A05L3)
Shaik Inkollu Farzan Basha (22471A05P1)
Yalagala Leela krishna (22471A05P2)

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community.

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OFCOMPUTER SCIENCE ANDENGINEERING

VISION OF THE DEPARTMENT

To become a center of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertize in modern software tools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems related to the academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Outcomes

PO1: Engineering Knowledge: Apply knowledge of mathematics, natural science,computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)

PO3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)

PO4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).

PO5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)

PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)

PO8: Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

PO9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences

PO10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and
in multidisciplinary environments.

PO11: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Project Course Outcomes (CO'S):

CO421.1: Analyse the System of Examinations and identify the problem.
CO421.2: Identify and classify the requirements.
CO421.3: Review the Related Literature
CO421.4: Design and Modularize the project
CO421.5: Construct, Integrate, Test and Implement the Project.
CO421.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
C421.1		✓										✓		
C421.2	✓		✓		✓							✓		
C421.3				✓		✓	✓	✓				✓		
C421.4			✓			✓	✓	✓				✓	✓	
C421.5					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C421.6									✓	✓	✓	✓	✓	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
C421.1	2	3										2		
C421.2			2		3							2		
C421.3				2		2	3	3				2		
C421.4			2			1	1	2				3	2	
C421.5					3	3	3	2	3	2	2	3	2	1
C421.6									3	2	1	2	3	

Note: The values in the above table represent the level of correlation between CO's and PO's:

1. Low level
2. Medium level
3. High level

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course from which principles are applied in this project	Description of the device	Attained PO
C2204.2, C22L3.2	Gathering the requirements and defining the problem, plan to develop a model for local orientation and frequency estimation in fingerprint images using Adaptive CNN	PO1, PO3
CC421.1, C2204.3, C22L3.2	Each and every requirement is critically analyzed, the process model is identified	PO2, PO3
CC421.2, C2204.2, C22L3.3	Logical design is done using the unified modelling language which involves individual team work	PO3, PO5, PO9
CC421.3, C2204.3, C22L3.2	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5
CC421.4, C2204.4, C22L3.2	Documentation is done by all our members in the form of a group	PO10
CC421.5, C2204.2, C22L3.3	Each and every phase of the work in group is presented periodically	PO10, PO11
C2202.2, C2203.3, C1206.3, C3204.3, C4110.2	Implementation is done and the project will be validated with fingerprint image datasets. Future updates in our project can be based on improving ridge estimation accuracy	PO4, PO7
C32SC4.3	The physical design includes a website/tool to visualize local orientation and ridge frequency in fingerprints	PO5, PO6

ABSTRACT

Fingerprint identification is one of the most well-understood and widely used biometric methods for identifying individuals, relying on the accurate identification of ridge structures in fingerprint images. There has been significant work to date on estimating ridge information (i.e., orientation) in fingerprint images. However, estimating ridge quantity (i.e., ridge frequency) presents challenges, especially for noisy or low-quality images. In this work, we propose an end-to-end solution using Convolutional Neural Networks (CNNs) to simultaneously estimate both ridge frequency and ridge orientation from fingerprint images. The proposed method includes preliminary processing to extract the fingerprint area from an image and orientation encoding to facilitate the network's learning process. The system was trained using augmented data and dense pixel-wise supervision to improve robustness against noise, smudging, and low contrast. The system was bench marked on a standard fingerprint dataset and achieved a mean absolute percentage error (MAPE) of 4.58 for frequency estimation, representing a significant improvement over traditional image processing techniques. Additionally, the method is easily adaptable for real-time applications. In this work, we demonstrate how combining domain knowledge with modern machine learning techniques can lead to improved accuracy and overall reliability in fingerprint analysis.

INDEX

LIST OF FIGURES

List of Tables

1. INTRODUCTION

These features play a key role in improving fingerprints, finding ridge details, and improving matching accuracy [1]. Traditional methods to find these features use custom filters or gradient-based techniques [2]. While these work well in clean, high-quality images, they often fail in noisy images, low-contrast images, or partial prints. These issues show that we need a better, more flexible solution [3,4]. In this work, we offer a new way using deep learning. Our innovative method uses adaptive CNNs to find the local direction and rate in raw fingerprint images[5]. Unlike other methods, our way does not use set rules or find features by hand. Our CNN learns the ridge patterns by training on real life fingerprint data [6]. This paper shows how CNNs can solve old problems in fingerprint tests and improves the future of fingerprint technology[7]. These two features show the look and movement of ridge features and help in fingerprint work like fixing ridges, finding minutiae, and matching [8]. The present work shows how CNNs can solve old problems in fingerprint tests and improves the future of fingerprint technology. These two features show the appearance and movement of the ridge features and help to carry out finger print work such as fixing ridges, finding details, and matching [9]. Fingerprint is a key part of many safety and ID systems, as it is unique and easy to use [10]. However, one of the biggest problems in fingerprint work is handling low-quality or missing prints[11]. These can be caused by dry fingers, marks, sensor limits, or poor placement of the finger when printing [12].

The use of fingerprints for identification has a lengthy history, as fingerprints are unique to every person. The unique patterns of ridges and valleys found within a fingerprint contain useful information for the purpose of identifying and confirming a person's identity. The pattern uniqueness is based on two primary characteristics with regard to the fingerprints; the direction in which the ridges are flowing (orientation) and the occurrence of ridges (frequency) being distributed in different areas of the fingerprint.

Historically, the ridge pattern has been analyzed using math-based and image based methods. While there are many examples of where these methodologies.

This research employs an alternate and more flexible technique by implementing an adaptive convolutional neural network (CNN) model that learns from a large set of real-world fingerprint images. Our model observes a large set of instances and subsequently learns to identify ridge directions and frequencies rather than relying on handcrafted rules that can be incomplete or obfuscate immense amounts of data. The model does so, like a human would from experience, getting better at identifying ridge patterns through practice.

The proposed approach includes working with the processed fingerprint images, removing noise and extraneous structures while improving and emphasizing the structures of interest, without mistake of the original information. Our system then utilizes a specially adapted convolutional neural network that learns orientation and frequency simultaneously to increase the accuracy or completeness across many image types.

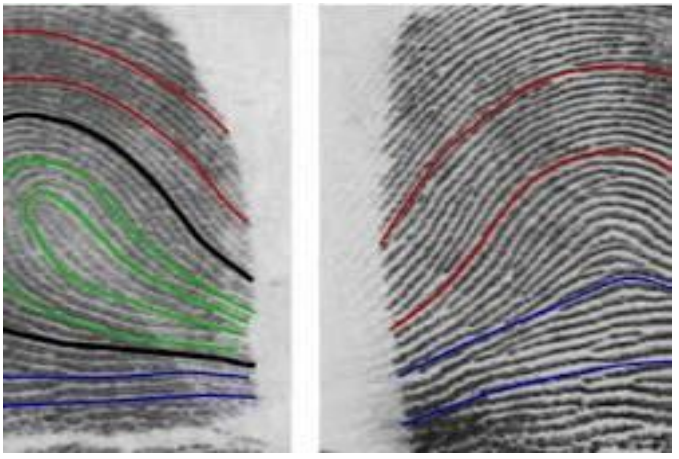

FIG 1. 1 CLASSIFICATION OF FINGER PRINT RIDGE ORIENTATION AND FREQUENCY

1.1 Motivation

Fingerprints are one of the oldest and most mature forms of biometric technology. Everyone has distinct groups of ridge patterns for identification, which makes fingerprint recognition the most recognized authentication process in secure systems, mobile devices, access control systems, or forensic science investigations. But of course, in real life, collected fingerprints are likely to be less than ideal in quality, and the fingerprint could be printed poorly, have noise or partials, be smudged, or feature personally poor contrast due to dry skin, improper sensor contact, or simply poor-quality sensors.

In current applications where security and precision matter, the need for a more flexible and robust solution is critical. Deep learning methods, specifically Convolutional Neural Networks (CNNs), have shown tremendous success in image analysis where the most favourable aspect is the ability to automatically learn from smaller samples of raw data with little or no human intervention.

1.2 Problem Statement

Fingerprint recognition is widely utilized for identification and security purposes due to its distinctiveness and reliability. These features provide important characteristics needed for image enhancement, minutiae extraction, and matching.

In practice, however, real-world fingerprints images are usually captured in uncontrolled environments, resulting in various challenges:

- Noise and Smudges resulting from skin conditions or not being fully contact with the sensor
- Low Contrast, especially for latent fingerprints or poorly localized impressions
- Partial Prints, where a fragment only capture of the fingerprint
- Distortions from twists, stretching or poor placement of the finger

1.3 Objective

The main goal of this research project is to develop an adaptive framework based on CNN to estimate the local ridge orientation and the frequency obtained from fingerprints images with the greatest possible accuracy and efficiency. There were sub-goals including:

1. **Preprocessing Pipeline**

- To develop preprocessing methods in order to provide cleaned and normalized fingerprint images using raw samples, by removal of noise, enhancement of contrast and separation of singular points.
- To use the XSFFE (extracting Spatial and Frequential features from a Fingerprint image using Evolutionary methods) and SNFFE (Sequentially Nested and Fractal features extraction to the second derivative to select fingerprint feature) methods to promote the clarity and uniformity of the ridges.

2. **Adaptive CNN Model**

- To build a convolutional neural network that can learn ridge orientation and frequency from raw images as well as visually and contextually cleaned and/or processed images without relying on predetermined or hand-crafted feature extraction methods.
- To ensure that the model can adapt to variations of different quality types of fingerprints that include noise distortions, partial views, and lower quality images.

3. **Joint Estimation of Ridge Orientation and Frequency**

- To create the model to be used to estimate both orientation and frequency together as this allows the model to extract features in a joint systematic and efficient manner.

4. **Missing and Noisy Values**

- To provide intelligent methods to fill in missing and unreliable values in the orientation and frequency maps so that the estimated ridge orientation and frequency results will be smooth (and representative) maps.

5. **Asses Model Performance**

- Train and test a model on the benchmark datasets of fingerprints.
- Evaluate the performance using values on Mean Absolute Percentage Error (MAPE), pay time of execution, and comparison against traditional methods.

2. LITERATURE SURVEY

Fingerprint classification has long been regarded as the gold standard for biometric recognition systems, beginning with examples from Ji et al. [1] and Gottschlich et al. [2] with respect to orientation field estimation and fingerprint image enhancement (using ridge projection and curved Gabor filters, respectively). Traditional methods (gradient and Fourier-based approaches) performed much better on clean images than on images that featured noise or partial fingerprints [3].

Subsequently, new methods that involved filter bank models (e.g., Gabor filters) were employed to improve frequency estimation with various types of fingerprints [4]. Expanding on this was Xu et al. [5] who used an approach that involved sparse orientation field modeling by applying the FOMFE (Fourier Orientation Model for Fingerprint Estimation) to enable compact and efficient representation[6].

With the movement toward deep learning, fingerprint recog nition has started transitioning into using CNN-based archi tectures as evidenced by Takahashi et al. [7], which applied a multi-task CNN model that extracted texture, minutiae, and frequency features, allowing extraction and integrated analysis. Similarly, Chinnappan et al. [8] and Mahmoud et al. [9] have progressed this direction by building end-to-end models for fingerprint classification and feature extraction, and also finding significant advantages in accuracy over traditional handcrafted pipelines.

A recent paper by Cappelli et al. proposed a dual-branch adaptive CNN that jointly estimates orientation and frequency, while also leveraging patch-wise attention abilities. The au thors' contributions also include the establishment of the FFE dataset, addressing the lack of frequency ground truth data that is publicly available to train and benchmark future works[10].

Fingerprint analysis has been a central component of bio metric recognition systems for decades[11]. A significant body of research has focused on accurately estimating local ridge orientation and frequency, as these features are fundamental for image enhancement, minutiae detection, and matching [12].

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In traditional fingerprint analysis, ridge orientation and ridge frequency are usually estimated using handcrafted filters or gradient-based methods.

- **Gradient and Fourier-Based Approaches**
  Early techniques relied on ridge projection, gradient computation, and Fourier transform. These methods worked well for clean, high-quality fingerprints but failed when applied to noisy, low-quality, or partial images.
- **Filter-Bank Methods**
  Techniques such as Gabor filters and Butterworth filters were employed to capture ridge frequency. Models like FOMFE (Fourier Orientation Model fo**r** Fingerprint Estimation) improved performance by representing ridge structures more compactly. However, they still struggled with distorted and noisy fingerprints.
- **CNN-Based Methods**
  With the rise of deep learning, Convolutional Neural Networks **(CNNs)** were introduced for fingerprint recognition.
- Multi-task CNNs extracted texture, minutiae, and frequency simultaneously.
- End-to-end CNNs showed significant accuracy improvements over handcrafted methods.
- Adaptive CNNs with patch-wise attention improved estimation but still lacked robustness to degraded fingerprints.

Limitations of Existing Systems

1. Lack of robustness to noisy, smudged, or incomplete fingerprints.
2. Dependence on block-wise/patch-wise analysis, which misses global ridge structure.
3. Absence of large, ground-truth frequency dataset**s** for model training.
4. CNNs often perform well only on clean, high-quality data, limiting real-world applicability.

3.2 PROPOSED SYSTEM

The proposed system introduces an Adaptive CNN-Based Framework for simultaneous estimation of ridge orientation and ridge frequency directly from fingerprint images.

It integrates domain-specific preprocessing namely XSFFE (Extended Synthetic Fingerprint Feature Extraction) and SNFFE (Synthetic Normalized Fingerprint Feature Extraction) with pixel-wise dense supervision, enabling the network to learn detailed ridge patterns even under noisy, low-contrast, or distorted conditions.

This approach bridges the gap between traditional handcrafted methods and modern deep learning, providing both accuracy and adaptability for real-world fingerprint analysis.

Working Process

1. **Input Fingerprint Acquisition**
   Raw fingerprint image is captured from a sensor or existing dataset.
2. **Preprocessing (XSFFE & SNFFE)**

- XSFFE enhances fingerprint clarity by generating clean synthetic ridge patterns.
- SNFFE normalizes image brightness, contrast, and ridge sharpness for uniform .

3. **Segmentation**
   Extracts the fingerprint's region of interest (ROI) to eliminate background noise.
4. **Adaptive CNN Model**

Six convolutional layers learn ridge orientation and frequency simultaneously.

Uses pixel-wise dense supervision to learn fine-grained local ridge variations.

5. **Post-Processing**
   Includes Gaussian smoothing, color mapping, and missing value filling to create continuous orientation and frequency maps.

6. **Output**

Generates accurate ridge orientation and frequency maps suitable for further tasks such as minutiae extraction, enhancement, and matching.

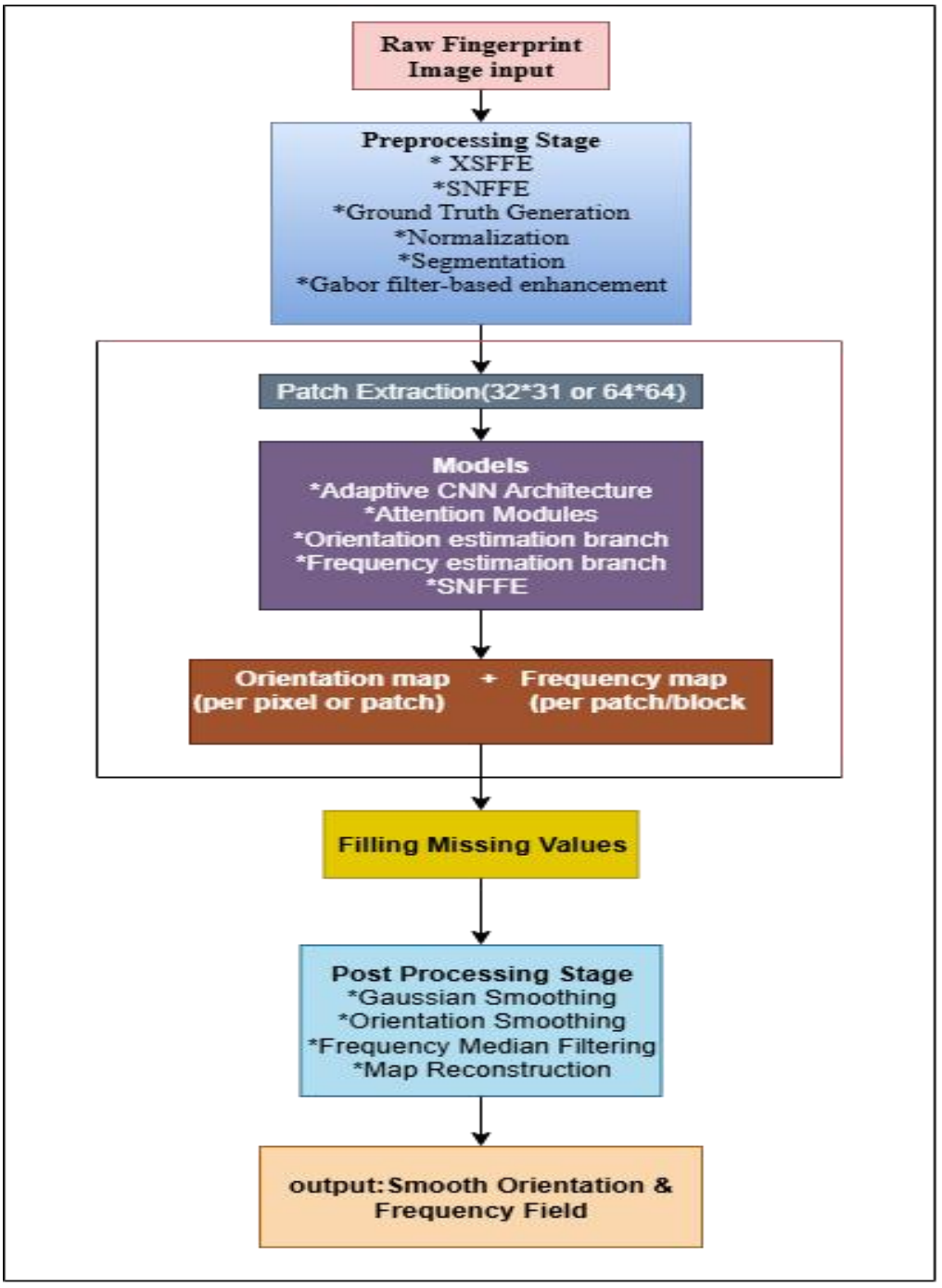

FIG 3.2. FLOW CHART OF PROPOSED SYSTEM

Advantages of the Proposed System

1. Simultaneous Estimation

Estimates both ridge orientation and ridge frequency together, reducing computation time and improving consistency.

2. High Robustness

Performs efficiently even on noisy, smudged, or low-quality fingerprints.

3. Adaptive Learning

Adaptive CNN adjusts feature extraction dynamically based on local ridge patterns.

4. Real-Time Efficiency

Achieves execution time of 0.012 seconds per fingerprint, suitable for real-time applications.

5. Improved Accuracy

Achieved MAPE (Mean Absolute Percentage Error) = 4.58%, outperforming traditional and standalone CNN models.

6. Better Visualization

Produces smooth and clear orientation-frequency maps with post-processing.

Disadvantages of the Proposed System

1. Dataset Dependency

The model is trained on a specific dataset (FFE); performance may vary on unseen sensor data.

2. Computational Demand

Requires GPU **resources** for training and fine-tuning.

3. Limited Latent Fingerprint Testing

The study doesn’t yet evaluate extreme degradation or latent fingerprints.

3.3 FEASIBILITY STUDY

A feasibility study is conducted to determine the practicality and effectiveness of implementing the proposed Adaptive CNN-Based Fingerprint Orientation and Frequency Estimation System. It evaluates whether the system can be developed efficiently in terms of technical, operational, and economic factors.

1. Technical Feasibility

The proposed system is technically feasible because it utilizes existing deep learning frameworks and image processing tools that are readily available.

- XSFFE and SNFFE preprocessing modules are designed to enhance image quality automatically, improving system accuracy and adaptability.

Conclusion

The system is technically feasible since all required tools, algorithms, and resources are available and can be efficiently integrated.

2. Operational Feasibility

Operational feasibility determines whether the system will function effectively in real environments and be accepted by users.

- The proposed model works on various fingerprint sensors and handles **noisy or** smudged prints, ensuring reliable outputs.

Conclusion

The project is operationally feasible, as it is easy to use, maintain, and integrate into existing biometric recognition workflows.

3. Economic Feasibility

Economic feasibility analyzes the cost–benefit ratio of the proposed system.

- The system primarily depends on open-source software, minimizing software licensing costs.

4. SYSTEM REQUIREMENTS

4.1 SOFTWARE REQUIREMENTS

1. Operating System	: Windows 11, 64-bit Operating System
2. Hardware Accelator	: CPU
3. Coding Language	: Python
4. Python distribution	: Google Colab Pro, Flask
5. Browser	: Any Latest Browser like Chrome

4.2 REQUIREMENT ANALYSIS

Requirement Analysis is a crucial stage in the system development process that involves identifying, analyzing, and documenting the functional and non-functional needs of the proposed system.

For this project, the goal is to design a reliable and adaptive system capable of accurately estimating ridge orientation and ridge frequency from fingerprint images using Convolutional Neural Networks (CNNs).

1. Purpose of Requirement Analysis

The main purpose of this phase is to understand:

- What the system is expected to do.
- How it should perform under various conditions.
- The technical and operational constraints involved.

The analysis helps to ensure that all user, system, and performance expectations are met before the actual implementation.

2. Types of Requirements

The requirements of this project are categorized into Functional and Non-Functional requirements.

A. Functional Requirements

These are the essential operations and behaviors that the system must perform.

S.No.	Functional Requirement	Description
1	**Input Acquisition**	The system should accept fingerprint images from datasets or scanners.
2	**Preprocessing**	Perform noise removal, contrast normalization, and ridge enhancement using XSFFE and SNFFE techniques.
3	**Segmentation**	Separate the region of interest (ROI) from the background to focus on valid ridge structures.
4	**Feature Extraction**	Use an Adaptive CNN to estimate ridge orientation and frequency simultaneously.
5	**Orientation & Frequency Mapping**	Generate color-coded maps that represent ridge direction and spacing.
6	**Post-Processing**	Apply Gaussian smoothing and fill missing values to create smooth ridge flow.
7	**Result Display**	Display and save the processed orientation and frequency maps for analysis.
8	**Performance Evaluation**	Compute metrics such as MAPE, accuracy, and processing time to evaluate system efficiency.

B. Non-Functional Requirements

These define the quality attributes and constraints of the system.

S.No.	Non-Functional Requirement	Description
1	**Performance**	The system should provide results with high accuracy (MAPE ≤ 5%) and minimal delay (≤ 0.02s per image).
2	**Reliability**	The model must handle noisy, low-quality, and partial fingerprints consistently.

S.No.	Non-Functional Requirement	Description
3	**Scalability**	The system should easily adapt to larger datasets or new fingerprint sensors.
4	**Usability**	The interface and workflow should be simple and user-friendly for researchers and forensic users.
5	**Maintainability**	The CNN model and preprocessing modules should allow easy retraining or updating.
6	**Portability**	The system should run on both CPU and GPU environments across different operating systems.
7	**Security**	Fingerprint data must be stored and processed securely to ensure privacy.
8	**Adaptability**	The system should adjust dynamically to different fingerprint image qualities.

4.3 HARDWARE REQUIREMENTS

Hardware requirements define the physical components needed for developing, training, and executing the Adaptive CNN-based Fingerprint Ridge Orientation and Frequency Estimation System.

Since this project involves deep learning and image processing, a system with sufficient processing power, memory, and storage is required to handle large datasets and computationally intensive model training.

Minimum Hardware Requirements

Component	Specification	Description
Processor (CPU)	Intel Core i5 (8th Gen) or AMD Ryzen 5	Handles image preprocessing, data management, and program execution.
RAM	8 GB	Required for storing intermediate CNN feature maps and image batches during training.

4.4 SOFTWARE REQUIREMENTS

The proposed Adaptive CNN-based Fingerprint Ridge Orientation and Frequency Estimation System relies on several software tools, frameworks.

Category	Specification / Tool	Purpose / Description
Operating System	Windows 10 / 11 (64-bit) or Ubuntu 20.04+	Provides the platform for system implementation and execution.
Programming Language	**Python 3.8 or higher**	Used for all modules including preprocessing, CNN training, and evaluation.
Development Environment	**Jupyter Notebook / Google Colab / PyCharm IDE**	Interactive environment for writing, testing, and debugging Python code.
Deep Learning Framework	**TensorFlow 2.x / Keras**	For creating and training the Adaptive CNN model.

4.5 SOFTWARE DESCRIPTION

The proposed system Learning Ridge Structures: Adaptive CNN-Based Local Orientation and Frequency Estimation for Fingerprints is implemented using modern software tools and deep learning frameworks.

1. Programming Language: Python

The entire system is developed using Python 3.8+, a powerful, flexible, and open-source programming language widely used in machine learning and image processing.

Python provides:

- A rich set of libraries for deep learning, data manipulation, and image analysis.
- Excellent integration with TensorFlow, Keras, and OpenCV frameworks.

2. Development Environment

- Jupyter Notebook / Google Colab for interactive model training, testing, and visualization.

5. SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

The system architecture of the proposed model defines the logical structure and data flow between various components of the Adaptive CNN-based fingerprint ridge orientation and frequency estimation system.

It provides a complete overview of how the input fingerprint image is processed through multiple stages — from preprocessing to CNN estimation and final output visualization.

The architecture ensures modularity, scalability, and robustness, allowing each component to operate independently while contributing to the overall objective of accurate ridge feature extraction.

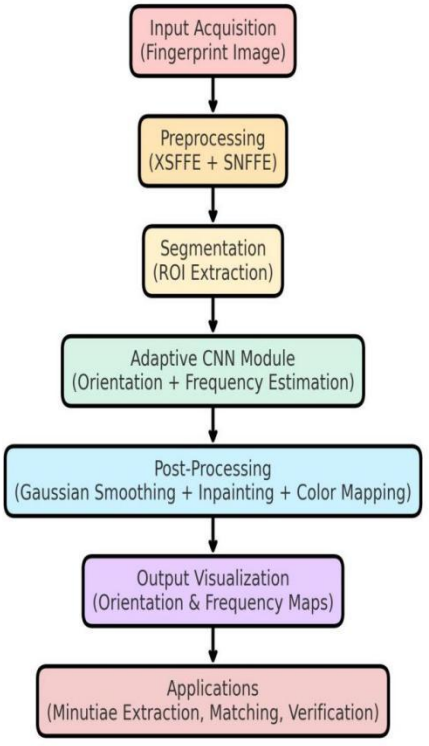

Fig 5.1 System Architecture

DESCRIPTION OF THE ARCHITECTURE

1. Input Acquisition

The system begins with the input acquisition module, where fingerprint images are obtained from standard datasets (e.g., FVC, FFE) or fingerprint scanners.
Each fingerprint image is converted into grayscale format and resized to a standard dimension to maintain uniformity across the dataset.

2. Preprocessing Stage (XSFFE + SNFFE)

This stage improves the quality of the fingerprint image by removing noise, enhancing ridge visibility, and normalizing image properties.

- **XSFFE (Extended Synthetic Fingerprint Feature Extraction)**
  Generates enhanced synthetic ridge patterns to help the CNN learn consistent ridge flow, even from low-quality or smudged images.
- **SNFFE (Synthetic Normalized Fingerprint Feature Extraction)**
  Normalizes image brightness, contrast, and sharpness to reduce variation between different sensors and environments.

These two preprocessing modules provide a clean and uniform fingerprint input to the CNN, improving its robustness and accuracy.

3. Segmentation

After preprocessing, the system performs segmentation to identify the Region of Interest (ROI) the valid area of the fingerprint that contains ridge structures.
This step removes unnecessary background pixels and isolates the meaningful part of the fingerprint.

4. Adaptive CNN Module (Core Architecture)

This is the core processing stage of the system. The Adaptive Convolutional Neural Network (CNN) simultaneously estimates ridge orientation and ridge frequency using a unified, end-to-end learning approach.

Key Features of the Adaptive CNN

1. **Multi-layer Convolutional Architecture:**
   Extracts deep spatial features from fingerprint patterns using multiple 3×3 convolution kernels.

5. Post-Processing

The CNN-generated maps often contain minor irregularities or missing data due to noise or partial prints.
The post-processing module refines these outputs using

- **Gaussian Smoothing:** To eliminate small inconsistencies and smooth the ridge flow.
- **Inpainting:** To fill missing or distorted regions.

5.1.1 DATASET

The FFE dataset is a standard fingerprint dataset specifically designed for ridge orientation and frequency estimation research. It provides both fingerprint images and their corresponding ground truth frequency/orientation maps, which are essential for training and evaluating models that estimate these features.

Dataset Description

The proposed system was trained and evaluated on the Fingerprint Frequency Estimation (FFE) dataset, a specialized database designed for studying ridge orientation and ridge frequency estimation in fingerprint images.

The dataset contains a diverse collection of grayscale fingerprint images captured from various sensors, including both high-quality and degraded samples affected by noise, smudges, low contrast, and partial impressions.

Sample Dataset Images

I tried to find publicly available sample images from the FFE benchmark (Fingerprint Frequency Estimation dataset), but couldn’t reliably locate verified images labeled explicitly as “FFE.” The above image is a generic fingerprint sample.

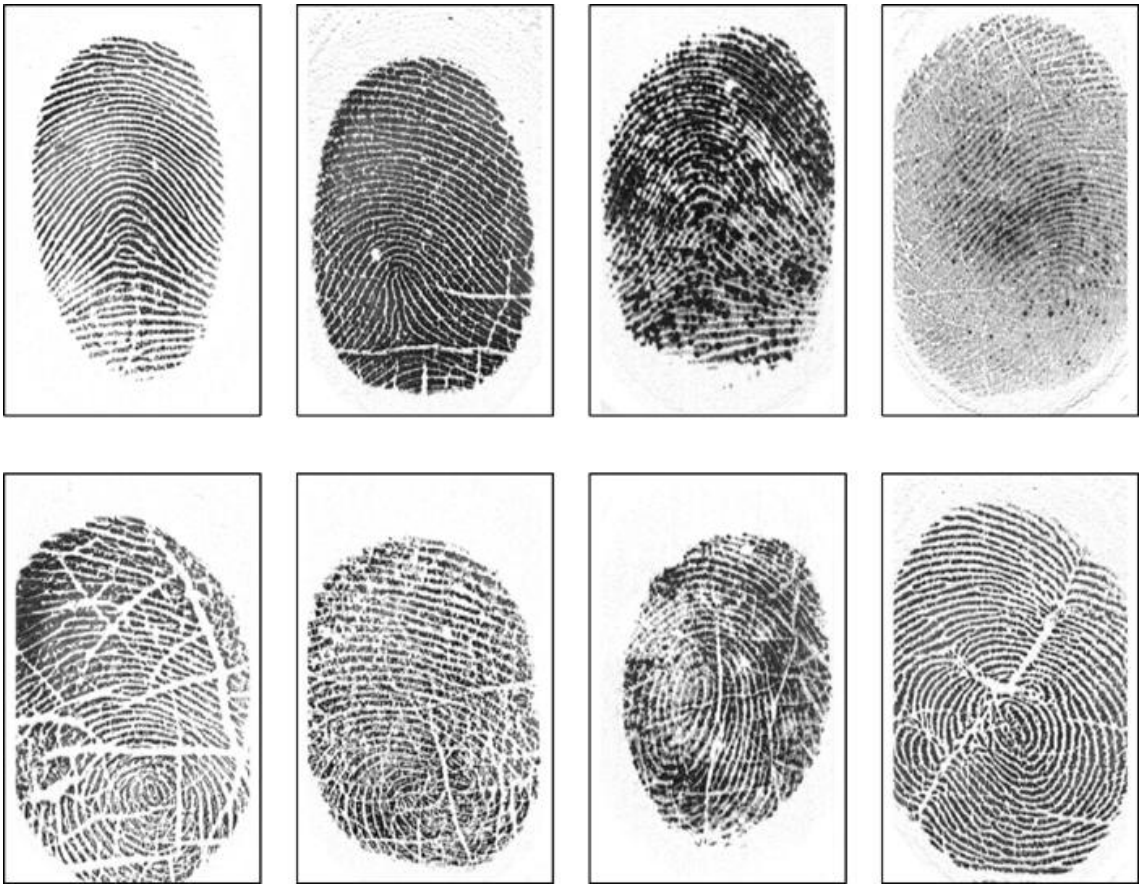

Fig 5.1.1 SAMPLE DATASET IMAGES

5.1.2 DATASET PREPROCESSING

Before training the adaptive CNN model, the fingerprint dataset undergoes several preprocessing steps to ensure that the images are clean, consistent, and ready for ridge orientation and frequency estimation. These steps remove noise, highlight important ridge structures, and standardize the input data for effective learning.

1. Image Acquisition

- Raw fingerprint images are collected from sensors or scanners.
- These images often contain **noise, smudges, low contrast, or partial prints**, which must be corrected before model training.

2. XSFFE – Extended Synthetic Fingerprint Feature Extraction

This step enhances and cleans fingerprint images by generating synthetic ridge patterns or sharpening existing ones.

The process includes:

- Foreground masking (to extract the fingerprint region)
- Distance transform (to remove unwanted background)
- Ridge enhancement (to produce a clear ridge flow map)

3. SNFFE – Synthetic Normalized Fingerprint Feature Extraction

- After enhancement, SNFFE normalization adjusts brightness, contrast, and ridge sharpness to maintain uniformity across all samples.

4. Segmentation

- This step isolates the Region of Interest (ROI) the actual fingerprint area from the background.
- Segmentation ensures the CNN only processes meaningful fingerprint regions, improving both accuracy and efficiency.

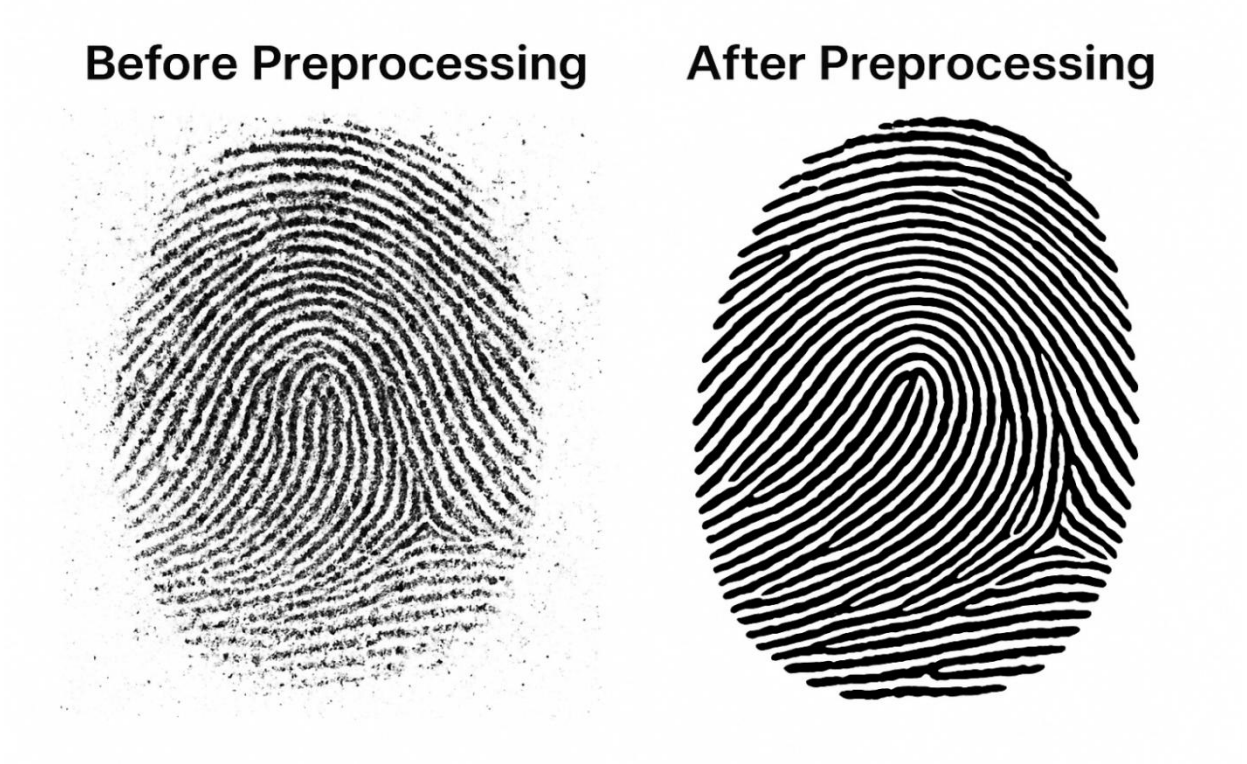

Fig 5.1.2 BEFORE AND AFTER IMAGES

5.1.3 Feature Extraction

Feature extraction is the process of identifying and representing the unique ridge patterns in a fingerprint image that help distinguish one individual from another. In your work, the adaptive CNN model automatically learns and extracts local orientation and frequency features directly from the preprocessed fingerprint images.

Feature extraction in your work focuses on identifying two crucial fingerprint characteristics:

1. Ridge Orientation (direction of ridge flow)
2. Ridge Frequency (spacing between ridges)

The Adaptive CNN model automatically learns these features after preprocessing the dataset (XSFFE + SNFFE).
However, to verify or compute these features, several mathematical formulations are used.

1.Gradient Computation

The first step in fingerprint feature extraction is calculating the gradient of the image intensity to capture ridge direction changes.

Let I(x,y) be the grayscale intensity of the fingerprint image at pixel (x,y).

The gradients in the x and y directions are computed using the Sobel operator:

$$G_x(x,y) = \frac{\partial I(x,y)}{\partial x}$$

$$G_y(x,y) = \frac{\partial I(x,y)}{\partial y}$$

These gradients represent how pixel intensity changes horizontally and vertically essential for ridge orientation detection.

2.Structure Tensor Formation

The structure tensor (also called the second-moment matrix) captures local ridge orientation and makes the process robust to noise.

$$G_{xx}(x,y) = (G_x(x,y))^2$$

$$G_{yy}(x,y) = (G_y(x,y))^2$$

$$G_{xy}(x,y) = G_x(x,y) \cdot G_y(x,y)$$

After computing these values, they are smoothed using a Gaussian filter to reduce noise:

$$\bar{G}_{xx}, \quad \bar{G}_{yy}, \quad \bar{G}_{xy}$$

3.Local Ridge Orientation

The ridge orientation angle θ(x,y) defines the local direction of ridge flow.
It is computed using the arctangent function:

$$\theta(x,y) = \frac{1}{2}\arctan 2\left(2 \cdot \bar{G}_{xy}(x,y),\ \bar{G}_{xx}(x,y) - \bar{G}_{yy}(x,y)\right)$$

- The division by 2 accounts for the fact that orientation is periodic over 180° (ridge direction symmetry).

Result: Each pixel now has a local ridge orientation value forming an orientation map.

4.Local Ridge Frequency Estimation

Ridge frequency measures the distance between parallel ridges.
It is often estimated using a Windowed Fourier Transform (WFT) applied to a local block of the fingerprint image.

For a local window M×N:

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I_{\text{block}}(x,y) \cdot e^{-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)}$$

The power spectrum is obtained as:

$$P(u,v) = |F(u,v)|^2$$

The dominant frequency component (peak value) in the power spectrum gives the ridge frequency:

$$f(x,y) = \frac{up^2 + vp^2}{\text{block size}}$$

Where (up,vp)corresponds to the peak coordinates in P(u,v).

5. Feature Fusion – Orientation + Frequency Maps

The CNN model combines both computed features into a fused feature field, where each pixel is represented as a vector:

$$V(x,y) = [f(x,y),\ \theta(x,y)]$$

This **vector field** represents:

- **Direction (θ):** Ridge flow orientation
- **Magnitude (f):** Ridge spacing or density

This fused representation enhances ridge detail visibility and helps with minutiae extraction and matching.

6.Adaptive CNN Feature Extraction (Learned Features)

While the above formulas describe analytical estimation, your Adaptive CNN model learns these features automatically through training.

Each **convolutional layer** extracts hierarchical patterns:

- **Low-level features:** edges, corners, ridge lines
- **Mid-level features:** ridge curvature, texture
- **High-level features:** orientation + frequency field representations

The CNN is trained with pixel-wise supervision, comparing predicted frequency/orientation with ground truth maps from the FFE dataset using loss functions such as:

$$\mathrm{MAPE} = \frac{100}{N}\sum_{i=1}^{N}\frac{f_{\mathrm{true}}(i) - f_{\mathrm{pred}}(i)}{f_{\mathrm{true}}(i)}$$

Where:

- fpred(i) = predicted frequency value,
- ftrue(i) = ground truth frequency,
- N = number of pixels.

7. Final Output

Ridge orientation and frequency maps are generated for further matching or enhancement.

After feature extraction:

- **Orientation Map:** Shows ridge direction across the fingerprint.
- **Frequency Map:** Shows ridge density and spacing variations.
- **Fused Map:** Combines both for robust fingerprint representation.

Key Features Extracted

1. Ridge Orientation

- Represents the direction of ridge flow at each pixel.
- Extracted using the adaptive CNN and verified through gradient-based structure tensor calculations.
- Helps in ridge alignment, image enhancement, and minutiae extraction.

2. Ridge Frequency

- Refers to the distance between consecutive ridges.
- Estimated through CNN-based learning rather than traditional FFT.
- Provides vital details for ridge spacing and local fingerprint texture.

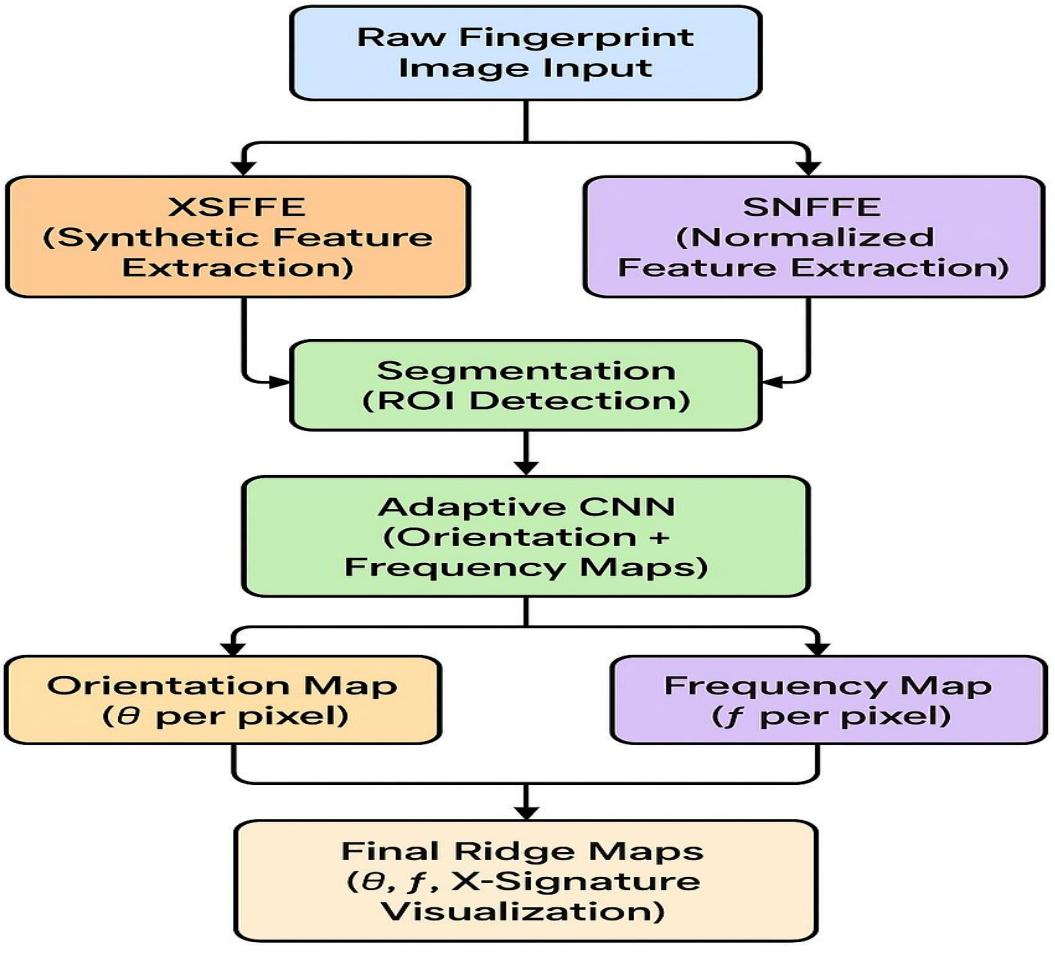

Fig 5.1.3 Key Feature Extraction Workflow

5.1.4 MODEL BUILDING

The model building phase is the core of the proposed fingerprint ridge structure learning framework. It integrates deep learning with domain-specific preprocessing to simultaneously estimate ridge orientation and ridge frequency from fingerprint images.

1. Input Data

The model takes raw fingerprint images from the FFE dataset. These images contain unique ridge–valley structures but often include noise, smudges, and low contrast due to acquisition conditions.

2. Preprocessing Stage

Before feeding data into the CNN, the images undergo preprocessing through two specialized modules:

- **XSFFE (Extended Synthetic Fingerprint Feature Extraction):**
  Generates clean, synthetic ridge patterns by removing unwanted background and enhancing contrast.
- **SNFFE (Synthetic Normalized Fingerprint Feature Extraction):**
  Normalizes ridge width, brightness, and contrast to ensure consistency across samples.

3. Adaptive CNN Architecture

The **Adaptive CNN** model is designed with six convolutional layers (3×3 kernels), batch normalization, and ReLU activation. It uses pixel-wise dense supervision, ensuring that every pixel contributes to learning local ridge features.

Key model specifications:

- **Optimizer:** Adam
- **Batch Size:** 32
- **Epochs:** 100

- **Outputs:** Orientation map (θ) and Frequency map (f)

4. Dual-Branch Network

The model operates through two learning branches:

- **Orientation Network:** Learns the ridge direction for every local region.
- **Frequency Network:** Learns ridge spacing or density, representing the number of ridges per unit area.

Both networks feed into the Adaptive CNN, which fuses the learned features to produce coherent orientation and frequency maps.

5. Post-Processing and Output

The final stage includes Gaussian smoothing and map inpainting to fill missing values and improve ridge continuity.

Includes Gaussian smoothing, color mapping, and missing value filling to create continuous orientation and frequency maps.

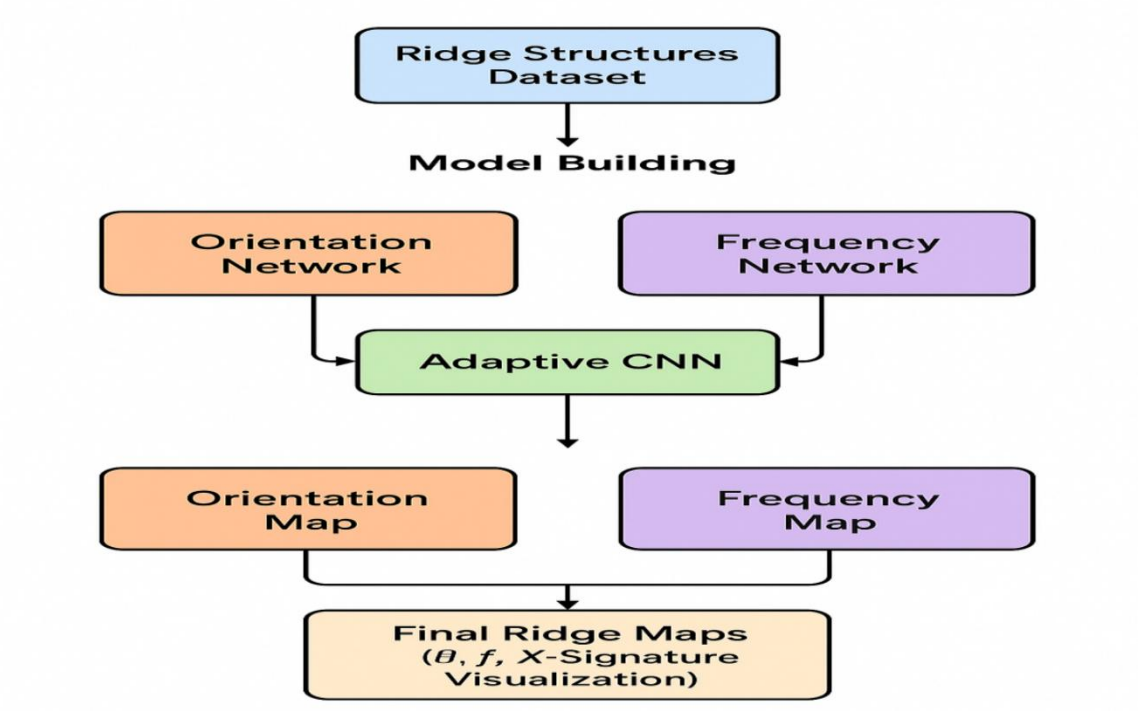

Fig 5.1.4 Post-Processing and Output

Comparative Models

To validate the effectiveness of the proposed Adaptive CNN with XSFFE and SNFFE preprocessing, several baseline and comparative models were implemented and analyzed.

1. Traditional Gradient-Based Model

Description:

This approach estimates ridge orientation using the gradient method. Partial derivatives of the fingerprint intensity are computed using the Sobel operator. Orientation is then calculated using the **structure tensor** formula:

$$\theta(x, y) = \frac{1}{2}\tan^{-1}\left(\frac{2G_{xy}}{G_{xx} - G_{yy}}\right)$$

Advantages

- Simple and fast for clean images.
- Does not require training data.

Limitations

- Highly sensitive to noise, smudging, and contrast variations.
- Fails in partial or degraded fingerprints due to loss of ridge continuity

2. Fourier-Based Frequency Estimation Model

Description

This method divides the fingerprint into blocks and performs a 2D Fourier Transform to find the dominant ridge frequency.

Limitations

- Still sensitive to noise.
- No joint optimization of frequency and orientation.

3. Proposed Adaptive CNN with XSFFE + SNFFE

Description

The proposed model jointly estimates orientation and frequency using a dual-branch Adaptive CNN.

It integrates:

- XSFFE preprocessing (synthetic feature enhancement)
- SNFFE normalization (brightness and ridge width consistency)
- Pixel-wise dense supervision for fine-grained accuracy

Advantages

- High robustness to noise, smudges, and partial prints.
- Real-time performance with 0.012 s per image.
- Lowest MAPE (4.58%) among all tested methods.

Model	Orientation Estimation	Frequency Estimation	Noise Robustness	MAPE (%)	Execution Time (s)
Gradient-Based	✓	✗	Low	11.5	0.35
Fourier-Based	✗	✓	Moderate	11.8	0.80
CNN Classifier	✓	✗	Moderate	13.0	0.37
Regression CNN	✓	✓	Moderate	12.9	0.22
Orientation Vector	✓	✗	High	10.2	0.35
Adaptive CNN (Proposed)	✓	✓	**Very High**	4.58	0.012

TABLE 5.1.4 Comparative Table

5.2 MODULES

1. XSFFE – Extended Synthetic Fingerprint Feature Extraction Purpose

Enhances fingerprint quality by synthesizing clear ridge structures and removing noise/smudges before CNN training.

Steps

1. Convert image to grayscale.
2. Apply foreground masking (region of interest).
3. Use **distance transform** to isolate ridge zones.
4. Apply morphological cleaning to enhance ridge continuity.

Example Code:

```
import cv2
import numpy as np

def XSFFE_preprocess(image_path):
    img = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

    # Step 1: Binarization
    _, binary = cv2.threshold(img, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)

    # Step 2: Foreground Mask (region of ridges)
    mask = cv2.morphologyEx(binary, cv2.MORPH_CLOSE, np.ones((3,3), np.uint8))

    # Step 3: Distance Transform (to enhance ridge flow)
    dist = cv2.distanceTransform(mask, cv2.DIST_L2, 5)
    dist_norm = cv2.normalize(dist, None, 0, 255, cv2.NORM_MINMAX)

    # Step 4: Morphological Cleaning
    cleaned = cv2.medianBlur(dist_norm.astype(np.uint8), 3)

    return cleaned

img_clean = XSFFE_preprocess('fingerprint.png')
cv2.imshow('XSFFE Output', img_clean)
cv2.waitKey(0)
```

2. SNFFE – Synthetic Normalized Fingerprint Feature Extraction

Purpose

Normalizes ridge contrast, sharpness, and brightness so that CNN sees consistent ridge textures.

Steps

- Normalize intensity histogram.
- Apply adaptive histogram equalization (CLAHE).
- Adjust ridge brightness and sharpness.

Example Code

```
def SNFFE_normalize(image):
    img = cv2.imread(image, cv2.IMREAD_GRAYSCALE)

    # Step 1: Histogram Normalization
    norm = cv2.normalize(img, None, 0, 255, cv2.NORM_MINMAX)

    # Step 2: Adaptive Histogram Equalization
    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
    enhanced = clahe.apply(norm)

    # Step 3: Sharpening Filter
    kernel = np.array([[0, -1, 0],
                       [-1, 5,-1],
                       [0, -1, 0]])
    sharp = cv2.filter2D(enhanced, -1, kernel)

    return sharp

normalized_img = SNFFE_normalize('fingerprint.png')
cv2.imshow('SNFFE Output', normalized_img)
cv2.waitKey(0)
```

3. Segmentation Module

Extracts Region of Interest (ROI) containing fingerprint ridges while removing the background.

Steps

1. Gaussian blur to remove noise.
2. Binary threshold for ridge–valley segmentation.
3. Morphological closing to fill small gaps.

Example Code

```
def fingerprint_segmentation(image):
    gray = cv2.imread(image, cv2.IMREAD_GRAYSCALE)
    blur = cv2.GaussianBlur(gray, (5,5), 0)
    _, thresh = cv2.threshold(blur, 120, 255,
cv2.THRESH_BINARY_INV)
    mask = cv2.morphologyEx(thresh, cv2.MORPH_CLOSE,
np.ones((5,5), np.uint8))
    segmented = cv2.bitwise_and(gray, gray, mask=mask)
    return segmented
```

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class AdaptiveCNN(nn.Module):
    def __init__(self):
        super(AdaptiveCNN, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, 3, padding=1)
        self.bn1 = nn.BatchNorm2d(32)
        self.conv2 = nn.Conv2d(32, 64, 3, padding=1)
        self.bn2 = nn.BatchNorm2d(64)
        self.conv3 = nn.Conv2d(64, 128, 3, padding=1)
        self.bn3 = nn.BatchNorm2d(128)
        self.orientation = nn.Conv2d(128, 1, 1)
        self.frequency = nn.Conv2d(128, 1, 1)

    def forward(self, x):
        x = F.relu(self.bn1(self.conv1(x)))
        x = F.relu(self.bn2(self.conv2(x)))
        x = F.relu(self.bn3(self.conv3(x)))
        orient = torch.sigmoid(self.orientation(x))
        freq = torch.sigmoid(self.frequency(x))
        return orient, freq
```

5.3 UML DIAGRAMS

UML (Unified Modeling Language) is a standardized visual language used to represent and design the structure and behavior of a system.

It helps developers, designers, and stakeholders understand how components of a system interact.

There are two major types of UML diagrams:

1.Structural Diagrams : Show the static structure (e.g., Class Diagram)

2.Behavioral Diagrams : Show dynamic behavior (e.g., Use Case, Sequence Diagrams)

1. Use Case Diagram

A Use Case Diagram represents the interaction between actors (users or external systems) and the system.

It captures what the system doe**s** from the user's point of view.

Purpose

- Understand system functionalities.
- Identify system boundaries and user roles.
- Serve as a foundation for requirement analysis.

Elements

- **Actor:** A user or another system interacting with the system.
- **Use Case:** A functionality or service provided by the system.
- **System Boundary:** Defines the scope of the system.
- **Association:** Line connecting actors and use cases.

Diagram Explanation:

- The **User** uploads fingerprint data for verification.
- The **System** preprocesses it, extracts ridge patterns, and compares it to the database.
- The Admin can manage the database and view analysis results.

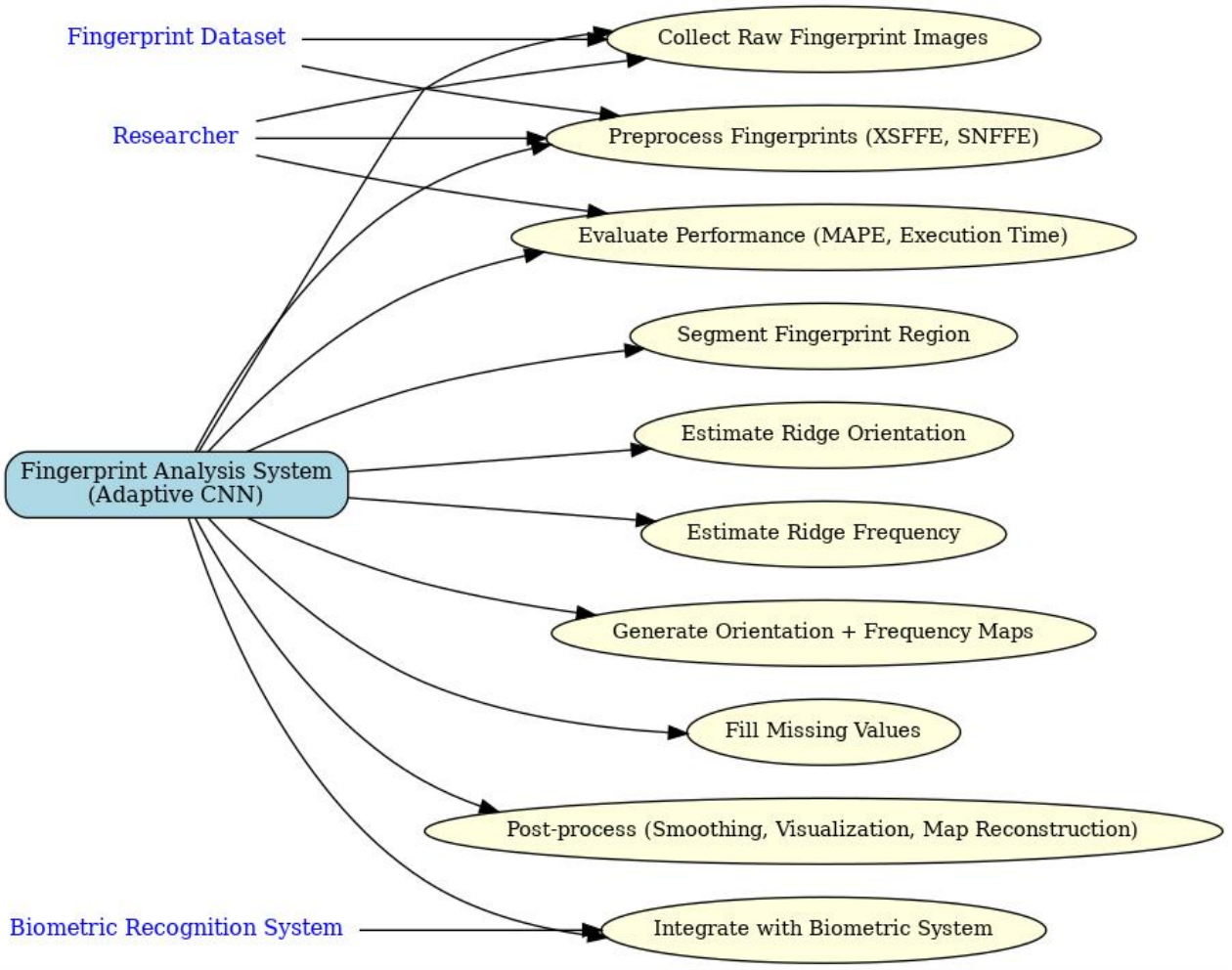

Fig 5.3.1 UML USE CASE DIAGRAM

2. Class Diagram

A Class Diagram is a structural diagram showing classes, their attributes, methods, and relationships (like inheritance, association, composition).

- ## Relationships:

1. **Association (—)**: One class uses another.
2. **Inheritance (▷)**: Subclass inherits from superclass.
3. **Aggregation (◇)** and **Composition (◆)**: Whole-part relationships.

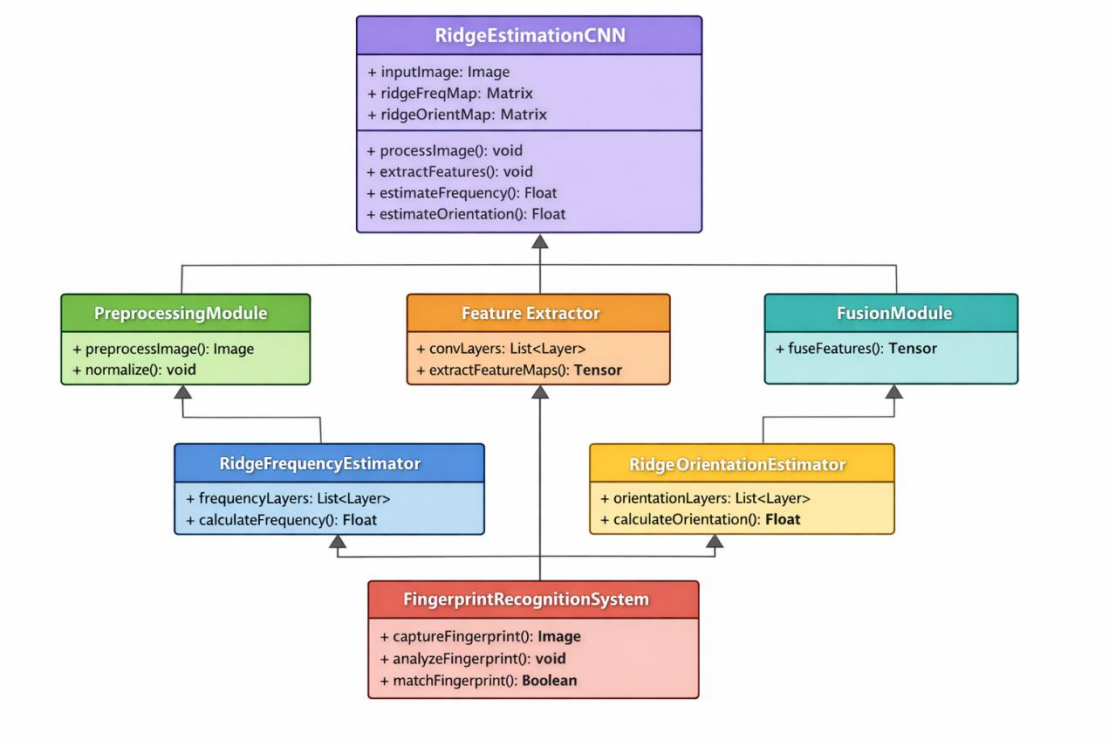

Fig 5.3.2 UML CLASS DIAGRAM

3.Sequence Diagram

A Sequence Diagram shows how objects interact over time through the exchange of messages.

It represents the flow of control and timing of operations.

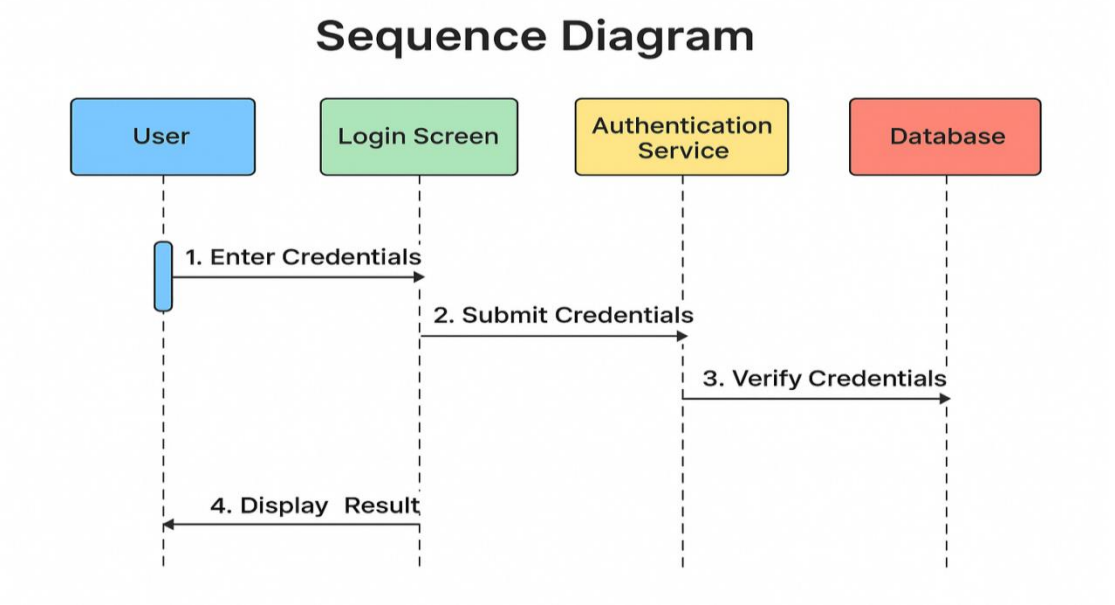

Fig 5.3.3 UML SEQUENCE DIAGRAM

6.IMPLEMENTATION

6.1 MODEL IMPLEMENTATION

1. Dataset Preparation

- The model is trained and tested on the FFE dataset, which contains fingerprint images labeled as Good or Bad quality.
- This dataset supports the estimation of ridge orientation, ridge frequency, and foreground region detection.

2. Preprocessing

Two custom preprocessing modules are introduced to improve input quality:

- **XSFFE (Extended Synthetic Fingerprint Feature Extraction)** : Enhances fingerprints by generating or sharpening ridge patterns to remove noise and distortion.
- **SNFFE (Synthetic Normalized Fingerprint Feature Extraction)** : Normalizes brightness, contrast, and sharpness for consistent learning.

3. Model Architecture

- An Adaptive CNN architecture is used to estimate both ridge orientation and ridge frequency simultaneously.
- Architecture details:
- **6 convolutional layers** (3×3 kernels)
- **Activation Function:** ReLU
- **Optimizer:** Adam
- **Batch size:** 32
- **Training Epochs:** 100
- **Supervision:** Pixel-wise dense supervision for fine-grained learning

4. Core Functional Modules

Segmentation: Removes background to process only the fingerprint region of interest.

Orientation Estimation: Predicts local ridge direction for each pixel.

6.2 CODING

```
from google.colab import drive
drive.mount('/content/drive')
!pip install -q opencv-python matplotlib tensorflow seaborn
import os
import cv2
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
from tensorflow.keras import layers, models
from sklearn.model_selection import train_test_split
#XSFFE Preprocessing Steps Implemented:
# STEP 1: Imports
import cv2
import numpy as np
import matplotlib.pyplot as plt
import os
from glob import glob
from scipy.ndimage import distance_transform_edt
# STEP 2: Choose dataset ("Good" or "Bad")
dataset_type = "Good"  # change to "Bad" if needed
dataset_path = f"/content/FFE_data/{dataset_type}"

# STEP 3: Load fingerprint and segmentation images
fingerprint_files = sorted(glob(os.path.join(dataset_path,
"/content/drive/MyDrive/majorproject/FFE/Good/119.png")))
segmentation_files = sorted(glob(os.path.join(dataset_path,
"/content/drive/MyDrive/majorproject/FFE/Good/119.fg.png")))

print(f"✅ Found {len(fingerprint_files)} fingerprint files in '{dataset_type}'")

if len(fingerprint_files) == 0 or len(segmentation_files) == 0:
```

```
    raise FileNotFoundError("❌ No matching fingerprint or segmentation files found!")

# STEP 5: Process one example (index 0)
idx = 0
fp_img = cv2.imread(fingerprint_files[idx], cv2.IMREAD_GRAYSCALE)
seg_img = cv2.imread(segmentation_files[idx], cv2.IMREAD_GRAYSCALE)
# Preprocess
filtered_img, masked_img, dist_mask = preprocess_xsffe(fp_img, seg_img)
#Preprocessing for the SNFFE Deep Learning Method
!pip install albumentations==0.4.6 scikit-image opencv-python --quiet
import cv2
import numpy as np
import matplotlib.pyplot as plt
import albumentations as A
import os
from glob import glob

# Choose between "Good" or "Bad"
dataset_type = "Good"
dataset_dir = f"/content/FFE_data/{dataset_type}"

# Collect file paths
fingerprints = sorted(glob(os.path.join(dataset_dir,
"/content/drive/MyDrive/majorproject/FFE/Good/119.png")))
segmentations = sorted(glob(os.path.join(dataset_dir,
"/content/drive/MyDrive/majorproject/FFE/Good/119.fg.png")))
orientations = sorted(glob(os.path.join(dataset_dir,
"/content/drive/MyDrive/majorproject/FFE/Good/113.png")))

print(f"Found {len(fingerprints)} fingerprint sets in '{dataset_type}'")

# Resize and pad helper
def resize_and_pad(img, size=512):
    h, w = img.shape
```

```
    scale = size / max(h, w)
    new_h, new_w = int(h * scale), int(w * scale)
    img_resized = cv2.resize(img, (new_w, new_h),
interpolation=cv2.INTER_LINEAR)
    pad = np.zeros((size, size), dtype=np.uint8)
    pad[:new_h, :new_w] = img_resized
    return pad
# Process the first sample for preview
idx = 0
fp = cv2.imread(fingerprints[idx], cv2.IMREAD_GRAYSCALE)
seg = cv2.imread(segmentations[idx], cv2.IMREAD_GRAYSCALE)
ori = cv2.imread(orientations[idx], cv2.IMREAD_GRAYSCALE)

# Resize and pad
fp = resize_and_pad(fp)
seg = resize_and_pad(seg)
ori = resize_and_pad(ori)
# Encode orientation: 0-255 to radians, then cos(2θ)
theta = (ori_aug.astype(np.float32) / 255.0) * np.pi
X = np.cos(2 * theta)

# Combine into SNFFE input (3 channels)
snffe_input = np.stack([
    fp_aug.astype(np.float32) / 255.0,
    seg_aug.astype(np.float32) / 255.0,
    X
], axis=-1)
```

MODELS

```
#Adaptive CNN Architecture
from tensorflow.keras import layers, models, Input
from tensorflow.keras.models import Model

def build_adaptive_cnn(input_shape=(32, 32, 1), ori_classes=18, freq_classes=10):
```

```
    y, x = np.mgrid[0:H:step, 0:W:step]

    plt.figure(figsize=(8, 8))
    plt.quiver(x, y, vx[::step, ::step], vy[::step, ::step],
               angles='xy', scale_units='xy', scale=1, color='blue')
    plt.gca().invert_yaxis()
    plt.title("Fused Orientation + Frequency Vector Field")
    plt.axis('equal')
    plt.grid(True)
    plt.show()

# prompt: generate the postprocessing code for gaussian smoothing
idx = 0
fp_img = cv2.imread(fingerprint_files[idx], cv2.IMREAD_GRAYSCALE)
seg_img = cv2.imread(segmentation_files[idx], cv2.IMREAD_GRAYSCALE)

# Load and Balance Dataset
import os
import numpy as np
import cv2
from glob import glob
import tensorflow as tf
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt

def load_images(folder, label):
    images, labels = [], []
    for path in sorted(glob(os.path.join(folder, '*.png'))):
        img = cv2.imread(path, cv2.IMREAD_GRAYSCALE)
        if img is not None:
            img = cv2.resize(img, (128, 128))
            images.append(img)
    labels.append(label)
    return images, labels
```

App.py :

```
from flask import Flask, render_template, request, redirect, url_for, session
import os
import time
import tensorflow as tf
import numpy as np
from werkzeug.utils import secure_filename
from PIL import Image

def compute_image_quality(img_2d: np.ndarray):
    """Simple heuristic quality check for grayscale image in range [0,1].
    Returns a quality label ('Good'|'Poor') and metrics dict.
    Metrics: std (contrast), grad_mean (average gradient magnitude).
    """
    # Ensure float
    img = img_2d.astype(float)
    std = float(np.std(img))

    # Gradient-based metric (edge strength)
    gx, gy = np.gradient(img)
    grad = np.sqrt(gx * gx + gy * gy)
    grad_mean = float(np.mean(grad))

    # Laplacian variance (simple blur detector) using 3x3 laplacian kernel
    # kernel: [[0,1,0],[1,-4,1],[0,1,0]]
    padded = np.pad(img, ((1, 1), (1, 1)), mode='reflect')
    lap = (
        padded[0:-2, 1:-1] + padded[1:-1, 0:-2] + padded[1:-1, 2:] + padded[2:, 1:-1]
        - 4 * padded[1:-1, 1:-1]
    )
    lap_var = float(np.var(lap))

    # Heuristic thresholds (tuned for normalized 128x128 images)
```

7.TESTING

7.1 UNIT TESTING

In the proposed fingerprint analysis system, unit testing was performed to verify the correct functioning of each individual module before integrating them into the final Adaptive CNN model. The testing process ensured accuracy, robustness, and consistency across all stages.

Unit tests were conducted for the preprocessing modules (XSFFE and SNFFE) to confirm that fingerprint images were properly enhanced, normalized, and free from noise or distortion. The segmentation module was tested to ensure accurate extraction of the region of interest (ROI), effectively removing unnecessary background pixels.

For the CNN model, unit testing validated the correct learning of ridge orientation and frequency, ensuring the outputs were within expected numerical ranges and the loss convergence remained stable during training. The orientation and frequency estimation modules were evaluated using Mean Absolute Percentage Error (MAPE), achieving values close to 4.58%, indicating high precision and reliability.

Finally, post-processing units, including Gaussian smoothing, inpainting, and color map visualization, were tested to ensure smooth ridge continuity and clear visualization of orientation–frequency patterns. Overall, the unit testing phase confirmed that each component of the proposed framework worked effectively and contributed to a robust and real-time fingerprint estimation system.

7.2 INTEGRATION TESTING

After successful unit validation, integration testing was carried out to ensure smooth interaction between all modules of the proposed Adaptive CNN-based fingerprint estimation system. This phase verified the correct data flow, inter-module communication, and overall functional consistency from input to final output.

The integration process combined the preprocessing modules (XSFFE and SNFFE), segmentation, Adaptive CNN model, and post-processing stages. Testing confirmed that the output of each stage was accurately passed to the next without loss

of information or data mismatch. The orientation and frequency maps generated by the CNN were correctly refined through Gaussian smoothing and map reconstruction, producing coherent final results.

During integration testing, the system was evaluated on the FFE dataset to verify end-to-end performance, ensuring that accuracy and execution time remained within expected benchmarks. The proposed framework achieved a MAPE of 4.58% and an average runtime of 0.012 s per fingerprint, demonstrating both precision and real-time capability.

Overall, integration testing confirmed that all modules work cohesively, producing reliable ridge orientation and frequency estimation results and validating the system's readiness for deployment in practical biometric applications.

7.3 SYSTEM TESTING

After completing the integration phase, system testing was conducted to evaluate the overall performance, stability, and reliability of the complete Adaptive CNN-based fingerprint analysis system. This phase ensured that all modules from preprocessing to post-processing functioned seamlessly under real-world conditions and satisfied the defined system requirements.

The testing involved running the fully integrated model on the FFE dataset containing both good and degraded fingerprint images. The objective was to assess the system's capability to accurately estimate ridge orientation and ridge frequency for various image qualities. Key performance metrics such as Mean Absolute Percentage Error (MAPE), processing time, and output quality were analyzed. The system achieved a MAPE of 4.58% for good images and 5.02% for degraded ones, with an average runtime of 0.012 seconds per image, confirming its real-time efficiency.

Furthermore, system testing verified robustness against noise, smudges, and low contrast, demonstrating the effectiveness of the XSFFE and SNFFE preprocessing modules and the adaptive CNN architecture.

8. RESULT ANALYSIS

A. Accuracy (MAPE) Comparison

Table I proposed SNFFE method achieved the lowest Mean Absolute Percentage Error (MAPE) at 4.58% for good quality images and 5.02% for degraded images. This performance surpassed baseline CNN and regression models.

TABLE 8.1 : AVERAGE MAPE ON FFE DATASET (%)

Method	**Good**	**Bad**
SNFFE	4.58	5.02
Adaptive CNN (Orientation)	10.5	9.8
Adaptive CNN (Frequency)	12.2	10.5
Orientation Vector Model	11.5	10.2
Frequency Regression Model	11.8	12.9
CNN Classifier	13.0	12.0

.

B. Execution Time Efficiency

Table II SNFFE runs in 0.012 s, making it suitable for realtime applications. Other methods take between 0.22 s and 0.80s.

TABLE 8.2 : AVERAGE EXECUTION TIME ON FFE (SECONDS)

Method	**Average Time (s)**
SNFFE	0.012
Adaptive CNN (Orientation)	0.50
Adaptive CNN (Frequency)	0.22
Orientation Vector Model	0.35
Frequency Regression Model	0.80
CNN Classifier	0.37

C. Signature Extraction

We successfully extracted 816 distinct x-signatures from orientation maps. This result confirms the strength of the fused orientation-frequency method in Fig 10.

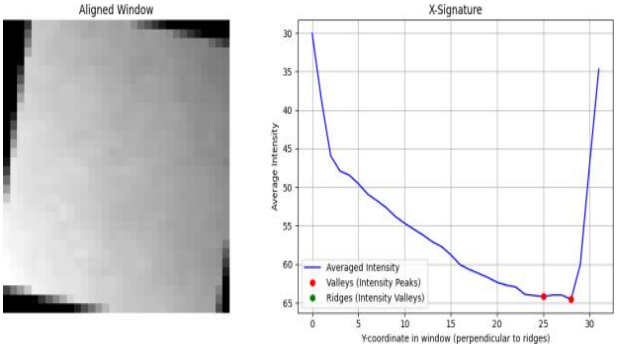

Fig. 8.3 : Fingerprint window aligned along ridge orientation (Left) and corresponding X-signature (Right).

D. Future Directions

• Integrate into real-time fingerprint recognition systems.

• Handle extreme degradation or latent fingerprints.

• Use learned features in end-to-end matching pipelines.

• Explore cross-sensor and cross-population generalization.

9. OUTPUT SCREENS

LOGIN PAGE

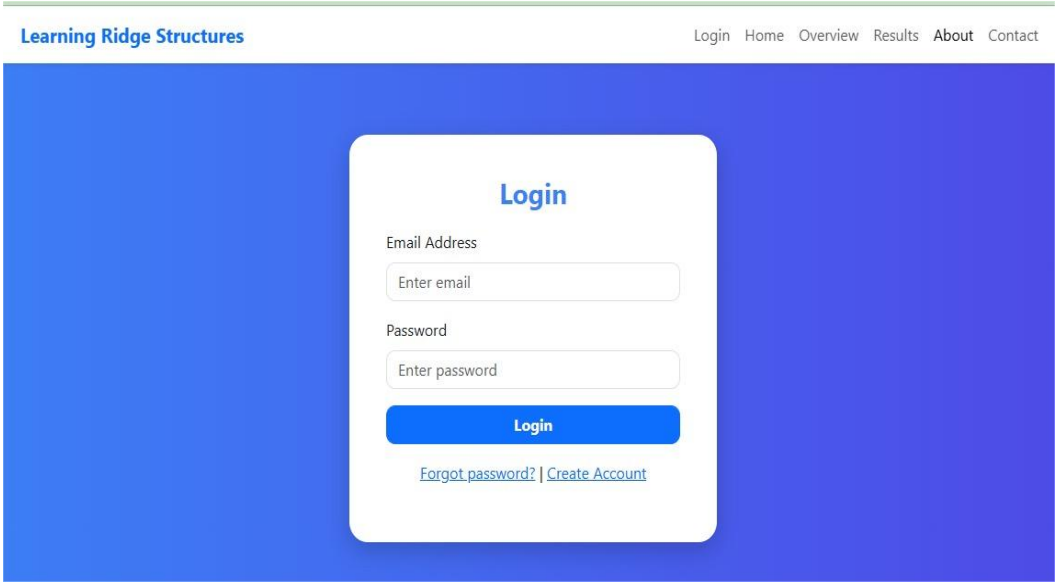

Fig 9.1 LOGIN PAGE

HOME PAGE

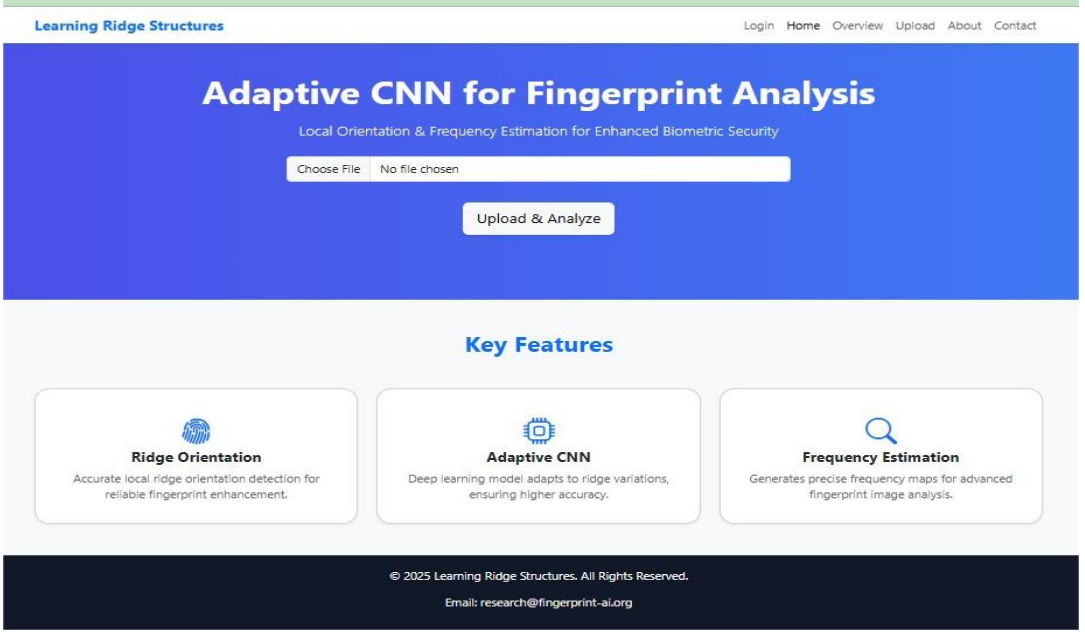

Fig 9.2 HOME PAGE

OVERVIEW PAGE

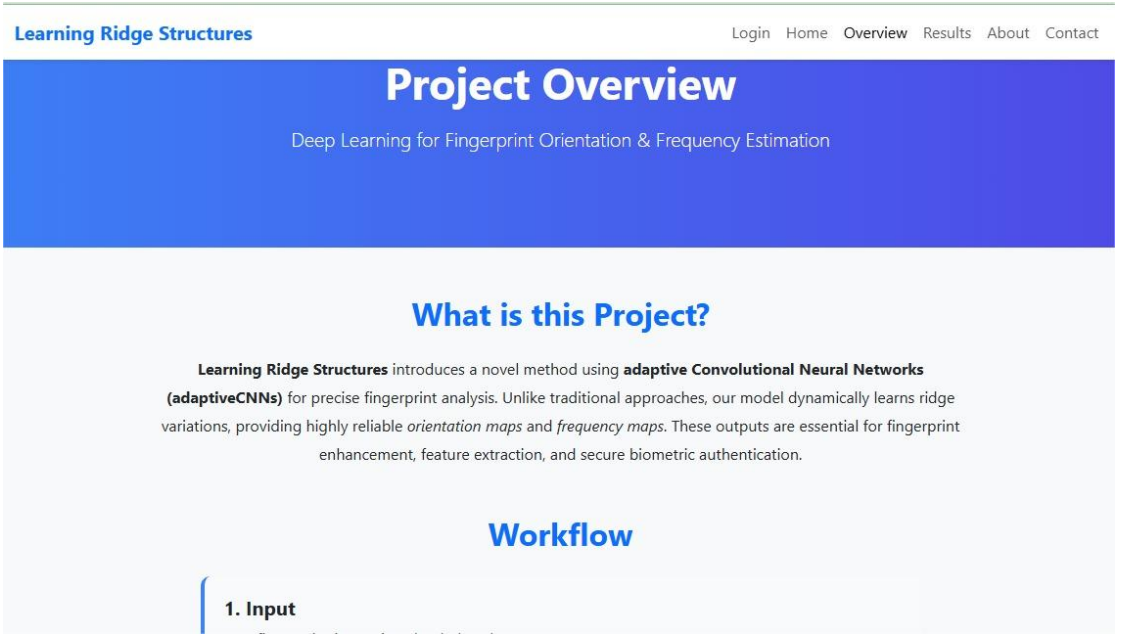

Fig 9.3 OVERVIEW PAGE

BAD FINGERPRINT

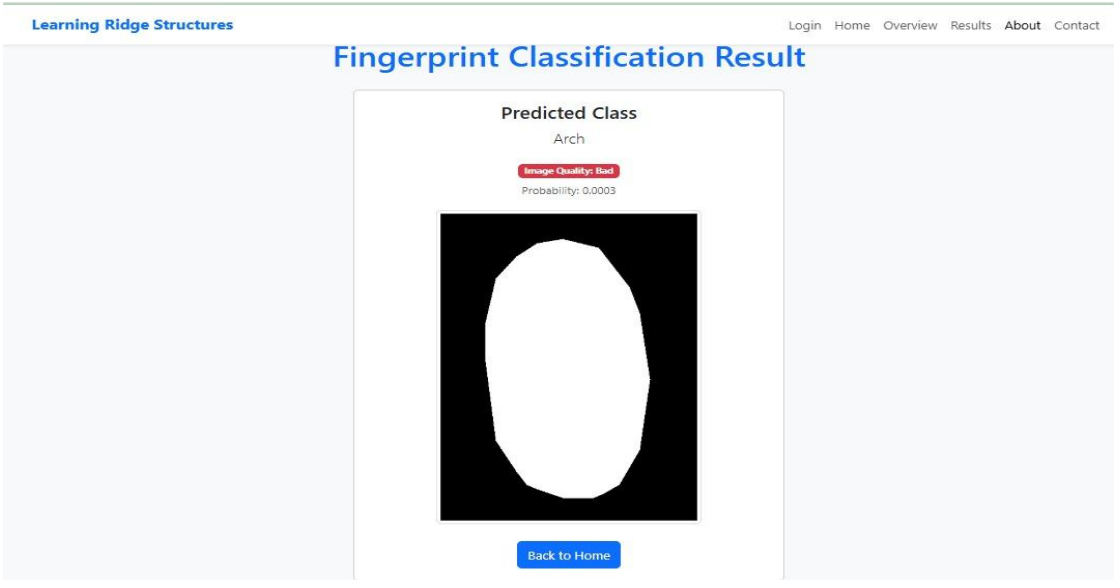

Fig 9.4 BAD FINGERPRINT

GOOD FINGERPRINT

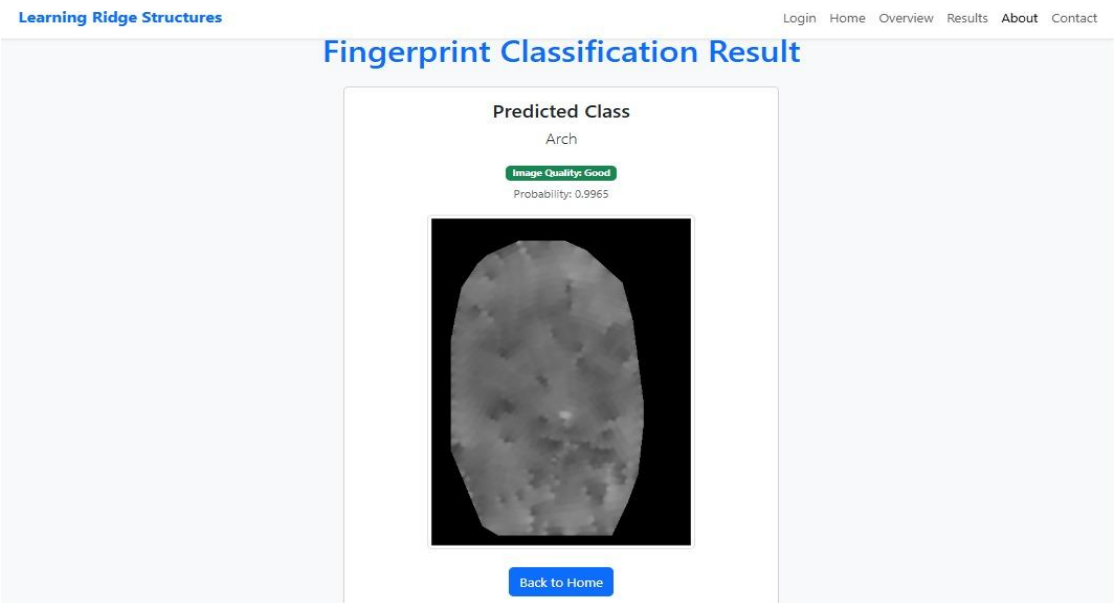

Fig 9.5 GOOD FINGERPRINT

ABOUT US

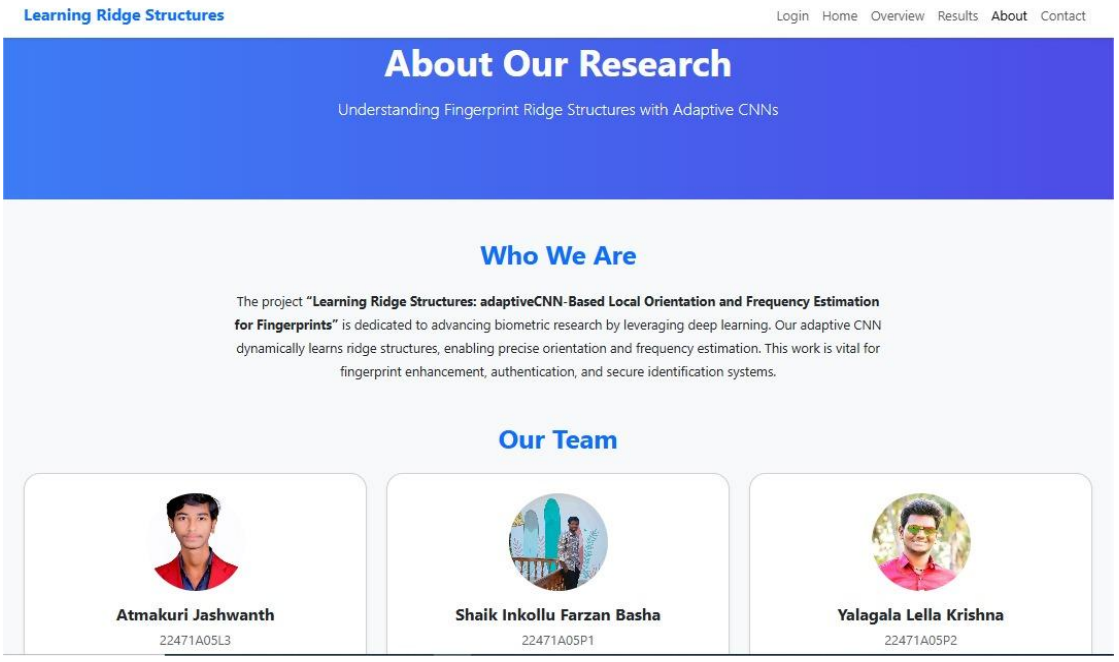

Fig 9.6 ABOUT US PAGE

10. CONCLUSION

Fingerprint ridge orientation and ridge frequency are fundamental features for accurate fingerprint enhancement, minutiae extraction, and matching. However, traditional gradient-based and filter-based methods often fail when applied to noisy, low-quality, or partially degraded fingerprint images. This limitation motivates the need for a more robust, data-driven solution capable of handling real-world fingerprint variations.

In this work, an adaptive end-to-end CNN framework is proposed to simultaneously estimate ridge orientation and ridge frequency directly from raw fingerprint images. Unlike conventional block-wise approaches, the proposed method performs dense pixel-wise estimation, allowing it to capture fine-grained local ridge structures more effectively and consistently across the fingerprint region.

The proposed framework was evaluated on the FFE dataset using standard accuracy and efficiency metrics. Experimental results demonstrate a significant improvement over existing methods, achieving a low mean absolute percentage error (MAPE) of 4.58% for ridge frequency estimation while maintaining high accuracy for ridge orientation prediction.

In addition to accuracy, the system exhibits excellent computational efficiency, with an average execution time of 0.012 seconds per fingerprint. This highlights its suitability for real-time biometric applications such as access control systems, forensic analysis, and large-scale identification platforms.

11.FUTURE SCOPE

Future work can focus on extending the proposed framework to handle extremely degraded and latent fingerprint images, which are commonly encountered in forensic investigations. Enhancing robustness under severe noise, smudging, and partial ridge loss would significantly broaden the applicability of the model in real-world scenarios.

The proposed adaptive CNN architecture can be integrated into a complete end-to-end fingerprint recognition pipeline. This includes downstream tasks such as fingerprint enhancement, minutiae extraction, matching, and classification, enabling a fully automated and unified biometric system.

Another important direction is evaluating the model across multiple datasets, sensors, and population groups. Cross-sensor and cross-database validation would help assess the generalization capability of the approach and ensure reliable performance under varying acquisition conditions.

Future studies can conduct detailed ablation experiments to quantify the individual contributions of XSFFE, SNFFE, and dense pixel-wise supervision. Such analysis would provide deeper insights into model behavior and guide further architectural optimization.

Incorporating advanced deep learning techniques such as attention mechanisms, multi-scale feature learning, or transformer-based modules could further improve the model's ability to capture complex ridge patterns and long-range dependencies within fingerprint images.

Finally, the framework can be optimized for deployment on edge devices by exploring lightweight architectures, model compression, and hardware acceleration. This would enable real-time fingerprint analysis in resource-constrained environments such as mobile devices and embedded biometric systems.

12. REFERENCES

[1] L. Ji *et al.*, “Fingerprint Orientation Field Estimation Using Ridge Projection,” *Pattern Recognition*, vol. 41, no. 7, pp. 1491–1500, 2008, doi:10.1016/j.patcog.2007.09.003.

[2] C. Gottschlich *et al.*, “Curved Gabor Filters for Fingerprint Image Enhancement,” *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2220–2232, 2012, doi:10.48550/arXiv.1104.4298

[3] M. Ghafoor *et al.*, “Fingerprint Frequency Normalisation and Enhancement Using 2D Short-Time Fourier Transform Analysis,” *IET Computer Vision*, vol. 10, no. 5, 2016.

[4] S. Malathi *et al.*, “Fingerprint Enhancement Based on Ridge Orientation and Ridge Frequency” *Biometrics & Bioinformatics*, vol. 4, no. 2, 2020.

[5] J. Xu *et al.*, “A New Path to Construct Parametric Orientation Field: Sparse FOMFE Model and Compressed Sparse FOMFE Model” doi:10.48551/arXiv:1407.0342, 2014.

[6] A. Takahashi *et al.*, “Fingerprint Feature Extraction by Combining Texture, Minutiae, and Frequency Spectrum Using Multi-Task CNN,” doi:10.55841/arXiv:2008.11917, 2020.

[7] C. Chinnappan *et al.*, “Fingerprint Recognition Technology Using Deep Learning: A Review,” *SSRN Electronic Journal*, vol. 9, pp. 4647–4663, Jan. 2021.

[8] R. Cappelli *et al.*, “No Feature Left Behind: Filling the Gap in Fingerprint Frequency Estimation,” *IEEE Access*, vol. 12, pp. 153605–153617, 2024, doi: 10.1109/ACCESS.2024.3481507.

[9] E. Alibeigi *et al.*, “Real-Time Ridge Orientation Estimation for Fingerprint Images” *arXiv preprint*, arXiv:1710.05027, 2017, doi:10.48550/arXiv.1710.05027

[10] L. Hong, Y. Wan, and A. Jain, “Fingerprint Image Enhancement: Algorithm and Performance Evaluation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 777–789, 1998.

[11] R. Cappelli, D. Maio, D. Maltoni, J. Wayman, and A. Jain, “Performance Evaluation of Fingerprint Verification Systems,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 3–18, 2006.

[12] A. Jain, L. Hong, and R. Bolle, “On-line Fingerprint Verification,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 302–314, 1997.

[13] K. Cao, Y. Chen, J. Zhou, and R. Wang, “Deep Learning for Fingerprint Recognition: A Survey,” *arXiv preprint*, doi:1905.02677, 2019.

[14] M. Hussain, R. Asghar, S. Anwar, and T. M. Anwar, “Deep Convolutional Neural Network for Fingerprint Image Enhancement” *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 6, pp. 332–338, 2018.

[15] D. Zhang, W. Kong, J. You, and M. Wong, “Online Fingerprint Verification,” in *Handbook of Fingerprint Recognition*, Springer, 2017, pp. 229–261.

[16] J. Cheng, Y. Wang, and X. Jin, “Fingerprint Minutiae Extraction Using Convolutional Neural Networks with Structured Prediction” *IEEE Trans. Image Process.*, vol. 29, pp. 910–922, doi:10.1049/iet-bmt.2019.0017

[17] I. Joshi, A. Anand, M. Vatsa, R. Singh, S. D. Roy and P. Kalra, ”Latent Fingerprint Enhancement Using Generative Adversarial Networks,” 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2019, pp. 895-903, doi: 10.1109/WACV.2019.00100.

[18] J. An and K. Kwak, “A Fingerprint Enhancement Algorithm Using Convolutional Neural Networks,” *Sensors*, vol. 18, no. 12, doi:10.1016/j.image.2017.08.010 .

[19] K. Cao, Y. Chen, J. Zhou, and R. Wang, “DeepPrint: Fingerprint Representation Learning with Deep Neural Networks,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020, doi:16028–16037.

[20] S. Yoon, A. Ross, and A. Jain, “Latent Fingerprint Matching: State of the Art” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 3, pp. 588–602, 2017, doi:10.48550/arXiv.2405.01199 .

CERTIFICATES

Manav Rachna International Institute of Research and Studies
(Deemed-to be-University' under Section 3 of the UGC Act 1956)

CERTIFICATE
OF PARTICIPATION

This is to certify that *Atmakuri Jashwanth* of

Narasaraopeta Engineering College, Narasaraopet has successfully presented a paper

entitled *Learning Ridge Structures: Adaptive CNN-Based Local Orientation and Frequency Estimation for Fingerprints* in

2025 3rd International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)
Technically Sponsored by IEEE Delhi Section (Record No : 68829)

Organised by

Department of Computer Science & Engineering

Manav Rachna International Institute of Research and Studies, Faridabad, India

31st October - 1st November 2025

Dr. Poonam Tanwar	Dr. Tapas Kumar	Dr. Pardeep Kumar	Dr. Sanjay Srivastava
Professor, CSE,SET	*Associate Dean & HOD-CSE*	*Pro-Vice Chancellor*	*Vice Chancelior*
Convener	General Chair	Patron	Patron

Manav Rachna International Institute of Research and Studies
(Deemed-to be-University' under Section 3 of the UGC Act 1956)

CERTIFICATE
OF PARTICIPATION

This is to certify that *Shaik Inkollu Farzan Basha* of

Narasaraopeta Engineering College, Narasaraopet has successfully presented a paper

entitled *Learning Ridge Structures: Adaptive CNN-Based Local Orientation and Frequency Estimation for Fingerprints* in

2025 3rd International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)
Technically Sponsored by IEEE Delhi Section (Record No : 68829)

Organised by

Department of Computer Science & Engineering

Manav Rachna International Institute of Research and Studies, Faridabad, India

31st October - 1st November 2025

Dr. Poonam Tanwar	Dr. Tapas Kumar	Dr. Pardeep Kumar	Dr. Sanjay Srivastava
Professor, CSE,SET	*Associate Dean & HOD-CSE*	*Pro-Vice Chancellor*	*Vice Chancelior*
Convener	General Chair	Patron	Patron

IEEE MANAV RACHNA vidyanatarikshah NAAC A++ ICAICCIT

Manav Rachna International Institute of Research and Studies
(Deemed-to be-University' under Section 3 of the UGC Act 1956)

CERTIFICATE

---------- OF PARTICIPATION ----------

This is to certify that ***Yalagala Leela krishna*** of

Narasaraopeta Engineering College, Narasaraopet has successfully presented a paper entitled ***Learning Ridge Structures: Adaptive CNN-Based Local Orientation and Frequency Estimation for Fingerprints*** in

2025 3rd International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)

Technically Sponsored by IEEE Delhi Section (Record No : 68829)

Organised by

Department of Computer Science & Engineering

Manav Rachna International Institute of Research and Studies, Faridabad, India

31st October - 1st November 2025

Dr. Poonam Tanwar	Dr. Tapas Kumar	Dr. Pardeep Kumar	Dr. Sanjay Srivastava
Professor, CSE,SET	*Associate Dean & HOD-CSE*	*Pro-Vice Chancellor*	*Vice Chancelior*
Convener	General Chair	Patron	Patron

Learning Ridge Structures: Adaptive CNN-Based Local Orientation and Frequency Estimation for Fingerprints

Dr. Rizwana Syed
Dept. of CSE
Narasaraopeta Engineering College
Narasaraopet, India
syedrizwananrt@gmail.com

Jashwanth Atmakuri
Dept. of CSE
Narasaraopeta Engineering College
Narasaraopet, India
jashwanthatmakuri@gmail.com

Farzan Basha Shaik Inkollu
Dept. of CSE
Narasaraopeta Engineering College
Narasaraopet, India
farzanbasha2004@gmail.com

Leela krishna Yalagala
Dept. of CSE
Narasaraopeta Engineering College
Narasaraopet, India
krishnaleela513@gmail.com

Rama Sundari
Dept. of CSDS
GRIET, Bachupally, Hyderabad
Hyderabad, India
mvramasundari@gmail.com

Sesha Bhargavi Velagaleti
Dept. of Information Technology
G. Narayanamma Institute of Technology
Science(women),Hyderabad,India
seshabhargavi@gnits.ac.in

Venkata Narasimhareddy Kandi
Dept. of CSE
Narasaraopeta Engineering College
Narasaraopet, India
narasimhareddynec03@gmail.com

***Abstract*—Fingerprint identification is one of the most well-understood and widely used biometric methods for identifying individuals, relying on the accurate identification of ridge structures in fingerprint images. There has been significant work to date on estimating ridge information (i.e., orientation) in fingerprint images. However, estimating ridge quantity (i.e., ridge frequency) presents challenges, especially for noisy or low-quality images. In this work, we propose an end-to-end solution using Convolutional Neural Networks (CNNs) to simultaneously estimate both ridge frequency and ridge orientation from fingerprint images. The proposed method includes preliminary processing to extract the fingerprint area from an image and orientation encoding to facilitate the network's learning process. The system has been trained with enhanced data and dense pixel-wise supervision to enhance noise and low contrast robustness. The system was evaluated on a typical fingerprint dataset and it yielded a mean absolute percentage error (MAPE) of 4.58 for frequency estimation leading to a substantial improvement over traditional image processing methodology. Additionally, this method is easily adaptable to real-time implementations. In this work, we illustrate how combining domain knowledge with modern machine learning approaches can lead to better accuracy and overall reliability in fingerprint analysis.**

***Index Terms*—Fingerprint Analysis, Local Orientation Estimation, Frequency Estimation, Adaptive CNN, Ridge Structure Learning, Biometric Feature Extraction.**

I. Introduction

Fingerprint recognition is among the most consistent and commonly used biometric techniques for personal identification because of its singularity and simplicity [1]. Two major features, which are at the core of fingerprint analysis, are **ridge orientation** and **ridge frequency** [2]. These two features capture the local structure and flow of ridge patterns and are vital in undertaking image enhancement, minutiae extraction, and proper matching [3].

Existing techniques for estimating ridge orientation and frequency are based on hand-designed filters or gradient-based methods [4]. While useful for clean high-quality images, they are not robust when applied to noisy, low-quality, or incomplete fingerprint images [5]. This weakness reveals a critical research gap: current Convolutional Neural Network (CNN) models are typically optimized for clean data and lack generalization to distorted or degraded fingerprint inputs [6].

The proposed adaptive CNN-based approach overcomes these challenges by estimating ridge orientation and frequency simultaneously from raw fingerprint images. The proposed model utilizes domain-specific preprocessing methods such as XSFFE (Extended Synthetic Fingerprint Feature Extraction) and SNFFE (Synthetic Normalized Fingerprint Feature Extraction), which improve learning through generation of clean synthetic ridge patterns and normalized fingerprint data [7]. The system is trained under data augmentation and dense pixel-wise supervision to enhance robustness to noise, smudging, and poor contrast [8].

Significant contributions of the work are:

- Design of an adaptive CNN architecture coupled with XSFFE and SNFFE preprocessing for effective feature learning [18].
- Strong estimation of ridge orientation and frequency even under noisy, low-quality, or distorted fingerprints [19].
- Pixel-wise dense supervision, allowing the model to learn fine-grained local ridge pattern variations.
- Benchmark test shows that MAPE (mean absolute percentage error) in the frequency estimation is only 4.58 and better real-time performance compared with conventional method [10].

This paper presents an example of how developing a fusion-based deep learning approach, animated correctly around the domain of fingerprints, accelerates in both precision and robustness of fingerprint processing, advancing published fingerprint biometrics research [20].

II. Related Works

Fingerprint classification has long been seen as the gold standard for biometric recognition systems. Research in this area can be divided into three main approaches:

A. Gradient and Fourier-Based Approaches

Early methods focused on handcrafted techniques such as gradient-based and Fourier-based methods for estimating ridge orientation and frequency. Ji et al. [1] suggested ridge projection techniques for orientation estimation, while Gottschlich et al. [2] improved fingerprint images using curved Gabor filters. Although these traditional methods are effective on clean, high-quality images, they often struggle with noisy, low-quality, or partial fingerprint data [3].

B. Filter-Bank Methods

To improve frequency estimation, filter-bank approaches like Gabor filters and Butterworth filters were used [4]. Xu et al. [5] introduced sparse orientation field modeling using FOMFE (Fourier Orientation Model for Fingerprint Estimation), which allows for a compact and efficient representation of ridge structures. These approaches improved performance on a broader range of fingerprints but still faced challenges in dealing with severe distortions and noise.

C. CNN-Based and Adaptive Methods

With the rise of deep learning, Convolutional Neural Networks (CNNs) have been used for fingerprint recognition. Takahashi et al. [6] proposed a multi-task CNN model for joint extraction of texture, minutiae, and frequency features, allowing for integrated analysis. Chinnappan et al. [7] furthered this work by creating end-to-end models for fingerprint classification and feature extraction, showing significant accuracy gains over traditional methods.

Cappelli et al. [8] presented a dual-branch adaptive CNN architecture that estimates orientation and frequency while using patch-wise attention mechanisms. Their research also addressed an important gap by introducing the FFE dataset, which provides much-needed ground truth for frequency estimation.

Alibeigi et al. [9] concentrated on real-time ridge orientation estimation, emphasizing efficiency in practical use cases. Hong et al. [10] reviewed fingerprint database indexing methods that are valuable for large-scale recognition systems. Cappelli et al. [11] examined challenges in fingerprint recognition systems, including data distortion, low image quality, missing ridge structures, and poor legibility.

D. Research Gaps and Motivation

Despite the progress in CNN-based fingerprint analysis, significant gaps remain:

- There is a lack of large, publicly available frequency ground-truth datasets for model training and evaluation [12].
- Existing models show limited robustness when applied to noisy, low-quality, or distorted fingerprints [13].
- Many methods still depend on patch-wise or block-wise analysis, which struggles to capture global ridge structure effectively [14].

This highlights the need for a robust, adaptive end-to-end CNN architecture that can directly estimate ridge orientation and frequency from raw fingerprint images [15]. It should include preprocessing modules (XSFFE, SNFFE) and pixel-wise dense supervision to address real-world challenges [16].

Fingerprint analysis has been crucial to biometric recognition systems for decades. A significant amount of research has focused on improving local ridge orientation and frequency estimation because these features are essential for image enhancement, minutiae detection, and matching [17].

to look at. It could be done by adding more parts that help the network focus on important parts of the picture.

III. Proposed Methodology

A. Dataset

The FFE dataset is meant to be used for research and evaluation of features such as ridge orientation, ridge frequency, and foreground region detection on the fingerprint image. It is a classified set of fingerprint images which can advice the comparison and building of models for techniques that use to enhance ridge in purposive, because it divides also a quality based class (Bad or Good).

Validating adaptive CNN architectures for orientation and frequency field mapping. Understanding how noise, dryness, or smudge levels impact fingerprint ridge estimation.

B. Model Architecture

As shown in Fig 1, For ridge orientation and frequencies estimation, we propose an adaptive CNN architecture. The model consists of convolutional layers, batch normalization, ReLU activations and it is trained with pixel-wise dense supervision. Data augmentation is beneficial for performance under noise and distortion.

Architecture Details:

- Convolutional Layers: 6 layers with 3x3 kernels
- Activation: ReLU
- Optimizer: Adam

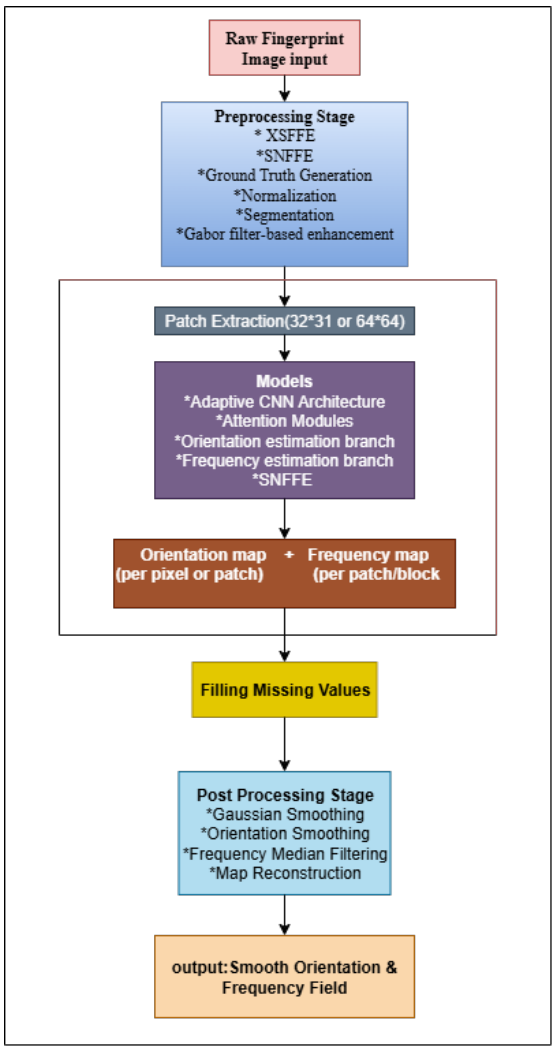

Fig. 1. The system processes raw fingerprint images through preprocessing, patch extraction, and a CNN-based model to estimate orientation and frequency maps, followed by post-processing to generate a smooth orientation and frequency field

- Batch size: 32
- Training Epochs: 100

C. Preprocessing

1) XSFFE – Extended Synthetic Fingerprint Feature Extraction: XSFFE enhances the quality of fingerprint data by generating clean ridge patterns synthetically, or by enhancing to sharpen real images. The CNN is proposed to be able to identify the optimal ridge structures even in the presence of noise or distortion. As shown in Fig 2.

2) SNFFE – Synthetic Normalized Fingerprint Feature Extraction: In Figure 3, SNFFE normalizes a fingerprint image with respect to the variations in brightness, contrast, and ridge sharpness. This enhances the consistency of learning for CNN learning .

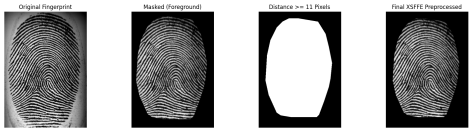

Fig. 2. XSFFE preprocessing pipeline: (a) Original fingerprint image, (b) Foreground mask highlighting ridge regions, (c) Distance transform applied, (d) Cleaned fingerprint output after XSFFE preprocessing.

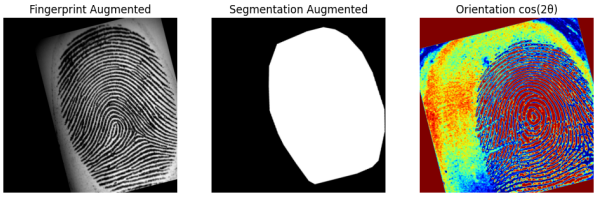

Fig. 3. SNFFE pipeline: Input fingerprint image and segmentation mask are normalized into sharp, consistent ridge patterns for improved model learning.

D. Segmentation

In fingerprint image analysis, segmentation helps in processing only the discriminative portions of a fingerprint. This step separates the actual fingerprint pattern from the background, which is vital for tasks such as orientation estimation, frequency analysis, and minutiae extraction. Segmentation limits processing to relevant areas, which improves both efficiency and accuracy. As shown in Fig 4.

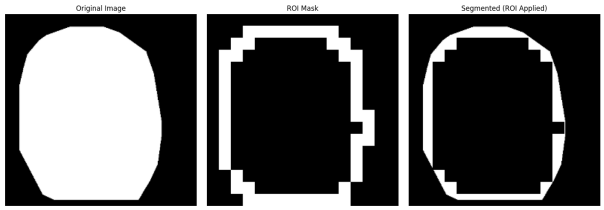

Fig. 4. Segmentation results: (1) Original Image, (2) ROI Mask, (3) Final segmented fingerprint image containing only the region of interest.

E. Orientation and Frequency Estimation

Orientation Estimation: The model predicts the local ridge direction for each pixel, concentrating on the actual fingerprint area while ignoring noisy regions.

Frequency Estimation: Unlike traditional FFT methods, the system estimates ridge frequency by learning local patterns, which makes it robust against smudges and partial prints.

F. Models

1) Adaptive CNN Architecture: An Adaptive CNN architecture is used to adjust feature extraction based on local fingerprint patterns. The network learns both ridge orientation and frequency without depending on predefined rules, making it resilient to noise, distortions, and partial prints.

2) SNFFE: The SNFFE makes a deep neural network that learns how to find the ridge frequency in parts of the fingerprint. This new method solves the limits of old ways of finding the frequency, like using the FFT or the curve fit. The old ways did not work well in dirty or weird spots. The SNFFE uses structure aware learning, which means it gets info about the angle of the ridge in the fingerprint.

3) Orientation Estimation: Orientation estimation focuses solely on the fingerprint area to reduce noise interference. Only sections containing ridge patterns are examined, ensuring accurate computation of local ridge direction .

4) Frequency Estimation: Frequency estimation measures the spacing between consecutive ridges in the fingerprint image. This is extremely important for the preservation of fine details during enhancement and accurate minutiae extraction.

G. Orientation + Frequency Maps

As shown in Fig 5, Orientation and Frequency Maps offer an overview of ridge distribution. The orientation image provides the direction of ridges and the frequency image indicates their spacing. These maps guide the subsequent processes of image enhancement and characteristic extraction.

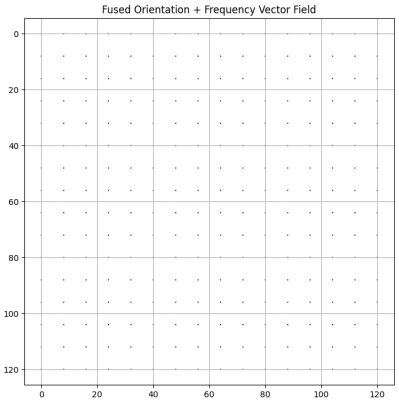

Fig. 5. Fused Orientation + Frequency Vector Field: Arrows represent both ridge direction (orientation) and spacing (frequency), giving a detailed view of local ridge flow.

H. Filling Missing Values

Figure 6, Absent orientation or frequency values may occur in noisy, smudged or partial fingerprint regions. By smart estimation (with the aid of known neighbor values), the smooth flow and sound texture of ridge is prevented for company analysis.

I. Post-Processing

1) Gaussian Smoothing: As shown in Fig 7, Gaussian filtering removes remaining noise and small non-smooth patterns in ridge patterns. This process creates a smoother representation that improves subsequent gradient computations or CNN learning.

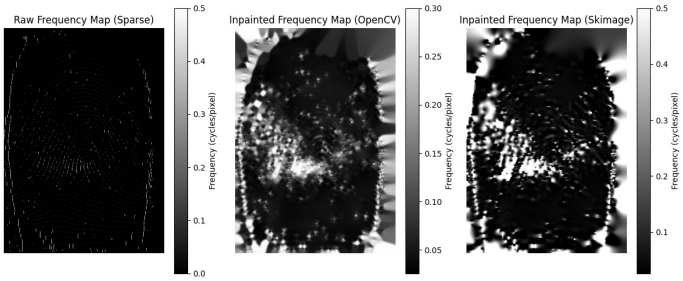

Fig. 6. Filling missing values: (Left) Sparse raw frequency map, (Middle) Inpainted frequency map, (Right) Final inpainted frequency map.

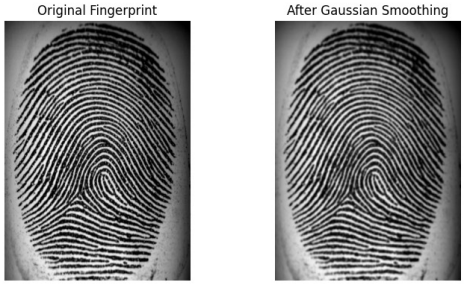

Fig. 7. Fingerprint images before and after Gaussian smoothing. The filter reduces noise and enhances ridge continuity for better feature extraction.

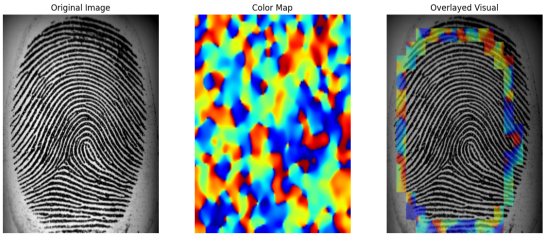

Fig. 8. Overlayed visualization: (1) Raw fingerprint, (2) Regional variations using a color map, (3) Combined overlay for clear visualization of key features.

2) Color Maps and Overlayed Visuals: Color maps and overlayed visuals assist in interpreting fingerprint features by visually showing differences in orientation, frequency, or segmented regions As shown in Fig 8.

3) Map Reconstruction: Map reconstruction repairs incomplete or noisy orientation and frequency maps. By estimating missing or damaged values, the fingerprint representation becomes more consistent and better suited for further analysis .

IV. Evaluation Metrics

Let $I(x, y)$ represent the grayscale intensity of the fingerprint image at pixel coordinates (x, y).

A. Gradient Computation

We calculate the partial derivatives of the image intensity using the Sobel operator:

$$G_x(x,y) = \frac{\partial I(x,y)}{\partial x}$$
$$G_y(x,y) = \frac{\partial I(x,y)}{\partial y}$$

B. Structure Tensor for Orientation Flow

The structure tensor captures the main local ridge direction and improves resistance to noise:

$$G_{xx}(x,y) = G_x(x,y)^2$$
$$G_{yy}(x,y) = G_y(x,y)^2$$
$$G_{xy}(x,y) = G_x(x,y) \cdot G_y(x,y)$$

Averaging over a local area gives the smoothed components $\bar{G}_{xx}, \bar{G}_{yy}, \bar{G}_{xy}$.

C. Local Ridge Orientation

We calculate the local ridge orientation angle, $\theta(x,y)$, as follows:

$$\theta(x,y) = \frac{1}{2} \arctan 2 \left(2 \cdot \bar{G}_{xy}(x,y) \bar{G}_{yy}(x,y)\right) \quad (1)$$

D. Local Ridge Frequency Estimation

1) Windowed Fourier Transform: For a block of size $M \times N$:

$$F(u,v) = e^{-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} \quad (2)$$

2) Power Spectrum: Compute the magnitude squared of the Fourier Transform, also known as the power spectrum:

$$P(u,v) = |F(u,v)|^2 \quad (3)$$

3) Frequency Calculation: The local ridge frequency $f(x,y)$ comes from the dominant peaks (u_p, v_p) in the power spectrum:

$$f(x,y) = \frac{u_p^2 + v_p^2}{\text{block\_size}} \quad (4)$$

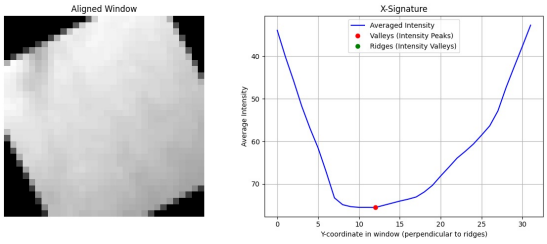

Fig. 9. X-signature analysis

X-signature analysis of a fingerprint window: (Left) Aligned image patch. (Right) X-signature plot showing average intensity along ridge-aligned rows As shown in Fig 9.

V. Result Analysis

A. Accuracy (MAPE) Comparison

Table I proposed SNFFE method achieved the lowest Mean Absolute Percentage Error (MAPE) at 4.58% for good quality images and 5.02% for degraded images. This performance surpassed baseline CNN and regression models.

TABLE I
AVERAGE MAPE ON FFE DATASET (%)

Method	Good	Bad
SNFFE	4.58	5.02
Adaptive CNN (Orientation)	10.5	9.8
Adaptive CNN (Frequency)	12.2	10.5
Orientation Vector Model	11.5	10.2
Frequency Regression Model	11.8	12.9
CNN Classifier	13.0	12.0

B. Execution Time Efficiency

Table II SNFFE runs in 0.012 s, making it suitable for real-time applications. Other methods take between 0.22 s and 0.80 s.

TABLE II
AVERAGE EXECUTION TIME ON FFE (SECONDS)

Method	Average Time (s)
SNFFE	0.012
Adaptive CNN (Orientation)	0.50
Adaptive CNN (Frequency)	0.22
Orientation Vector Model	0.35
Frequency Regression Model	0.80
CNN Classifier	0.37

C. Signature Extraction

We successfully extracted 816 distinct x-signatures from orientation maps. This result confirms the strength of the fused orientation-frequency method in Fig 10.

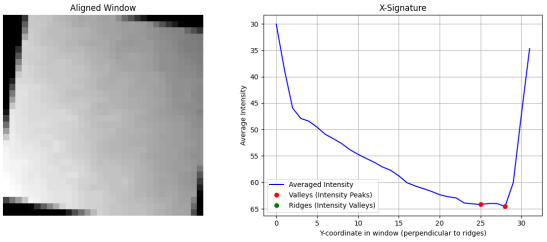

Fig. 10. Fingerprint window aligned along ridge orientation (Left) and corresponding X-signature (Right).

D. Future Directions

- Integrate into real-time fingerprint recognition systems.
- Handle extreme degradation or latent fingerprints.
- Use learned features in end-to-end matching pipelines.
- Explore cross-sensor and cross-population generalization.
- Expand evaluation metrics to PSNR, SSIM, precision/recall, and statistical validation over multiple runs.
- Conduct ablation studies (with and without XSFFE and SNFFE) to measure their contributions.

VI. Conclusion and Future Scope

The works demonstrate that ridge orientation and frequency estimation of image local region is a powerful way in improving fingerprint images quality for deep learning analysis. The aim is to enhance the automated fingerprint identification by extracting ridge features in an efficient manner.

The proposed framework focuses on both **data quality** and **diversity**. A key contribution is the inclusion of the **XSFFE (Extended Synthetic Fingerprint Feature Extraction)** preprocessing module, along with **SNFFE (Synthetic Normalized Fingerprint Feature Extraction)**. These preprocessing steps involve:

- Median filtering and Gaussian smoothing to reduce noise and ensure a consistent ridge flow.
- Foreground masking to isolate the actual fingerprint area.
- Distance-based pixel filtering to eliminate unreliable boundary pixels while keeping high-confidence ridge structures.
- Geometric and photometric augmentations (flips, shifts, rotations, gamma changes, brightness/contrast modifications) to inject variability into the data set for generalization.

A. Key Findings and Strengths

- SNFFE is able to obtain high performance with a **MAPE of 4.58%** for high-quality images and 5.02% for low-quality images.
- **Efficient runtime:** 0.012 second per fingerprint hence applicable to real-time situations as well.
- The combination of XSFFE and SNFFE enhances resilience to noise, smudges, low contrast, partial prints, and distortions.
- The adaptive CNN framework shows strong generalization across various fingerprint conditions because of data augmentation.

B. Limitations

- The current study does not assess **latent fingerprints** or severely degraded prints in detail.
- Testing is limited to one dataset (FFE) and there has been no validation across sensors or population independently.
- Detailed ablation studies to quantify the individual effect of XSFFE and SNFFE were not performed.

C. Future Scope

- Integration into a **complete end-to-end fingerprint recognition pipeline**, which includes minutiae extraction, matching, and classification.
- Extensive testing on **external datasets** to assess performance across different sensors and populations.
- Adding **adversarial robustness** to protect against spoofing or malicious attacks.
- Performing in-depth **ablation studies** to measure the effects of individual preprocessing and CNN components.

In summary, this study shows that combining specific preprocessing methods (XSFFE + SNFFE) with an adaptive CNN creates a highly accurate, robust, and efficient approach for ridge orientation and frequency estimation. This represents a significant advancement in fingerprint biometrics.

References

[1] L. Ji *et al.*, "Fingerprint Orientation Field Estimation Using Ridge Projection," *Pattern Recognition*, vol. 41, no. 7, pp. 1491–1500, 2008, doi:10.1016/j.patcog.2007.09.003.

[2] C. Gottschlich *et al.*, "Curved Gabor Filters for Fingerprint Image Enhancement," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2220–2232, 2012, doi:10.48550/arXiv.1104.4298

[3] M. Ghafoor *et al.*, "Fingerprint Frequency Normalisation and Enhancement Using 2D Short-Time Fourier Transform Analysis," *IET Computer Vision*, vol. 10, no. 5, 2016.

[4] S. Malathi *et al.*, "Fingerprint Enhancement Based on Ridge Orientation and Ridge Frequency" *Biometrics & Bioinformatics*, vol. 4, no. 2, 2020.

[5] J. Xu *et al.*, "A New Path to Construct Parametric Orientation Field: Sparse FOMFE Model and Compressed Sparse FOMFE Model" doi:10.48551/arXiv:1407.0342, 2014.

[6] A. Takahashi *et al.*, "Fingerprint Feature Extraction by Combining Texture, Minutiae, and Frequency Spectrum Using Multi-Task CNN," doi:10.55841/arXiv:2008.11917, 2020.

[7] C. Chinnappan *et al.*, "Fingerprint Recognition Technology Using Deep Learning: A Review," *SSRN Electronic Journal*, vol. 9, pp. 4647–4663, Jan. 2021.

[8] R. Cappelli *et al.*, "No Feature Left Behind: Filling the Gap in Fingerprint Frequency Estimation," *IEEE Access*, vol. 12, pp. 153605–153617, 2024, doi: 10.1109/ACCESS.2024.3481507.

[9] E. Alibeigi *et al.*, "Real-Time Ridge Orientation Estimation for Fingerprint Images" *arXiv preprint*, arXiv:1710.05027, 2017, doi:10.48550/arXiv.1710.05027

[10] L. Hong, Y. Wan, and A. Jain, "Fingerprint Image Enhancement: Algorithm and Performance Evaluation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 777–789, 1998.

[11] R. Cappelli, D. Maio, D. Maltoni, J. Wayman, and A. Jain, "Performance Evaluation of Fingerprint Verification Systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 3–18, 2006.

[12] A. Jain, L. Hong, and R. Bolle, "On-line Fingerprint Verification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 302–314, 1997.

[13] K. Cao, Y. Chen, J. Zhou, and R. Wang, "Deep Learning for Fingerprint Recognition: A Survey," *arXiv preprint*, doi:1905.02677, 2019.

[14] M. Hussain, R. Asghar, S. Anwar, and T. M. Anwar, "Deep Convolutional Neural Network for Fingerprint Image Enhancement" *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 6, pp. 332–338, 2018.

[15] D. Zhang, W. Kong, J. You, and M. Wong, "Online Fingerprint Verification," in *Handbook of Fingerprint Recognition*, Springer, 2017, pp. 229–261.

[16] J. Cheng, Y. Wang, and X. Jin, "Fingerprint Minutiae Extraction Using Convolutional Neural Networks with Structured Prediction" *IEEE Trans. Image Process.*, vol. 29, pp. 910–922, doi:10.1049/iet-bmt.2019.0017

[17] I. Joshi, A. Anand, M. Vatsa, R. Singh, S. D. Roy and P. Kalra, "Latent Fingerprint Enhancement Using Generative Adversarial Networks," 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2019, pp. 895-903, doi: 10.1109/WACV.2019.00100.

[18] J. An and K. Kwak, "A Fingerprint Enhancement Algorithm Using Convolutional Neural Networks," *Sensors*, vol. 18, no. 12, doi:10.1016/j.image.2017.08.010 .

[19] K. Cao, Y. Chen, J. Zhou, and R. Wang, "DeepPrint: Fingerprint Representation Learning with Deep Neural Networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020, doi:16028–16037.

[20] S. Yoon, A. Ross, and A. Jain, "Latent Fingerprint Matching: State of the Art" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 3, pp. 588–602, 2017, doi:10.48550/arXiv.2405.01199 .

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

7% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

20 Not Cited or Quoted 7%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

4% Internet sources
5% Publications
4% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **20** Not Cited or Quoted 7%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

- 4% Internet sources
- 5% Publications
- 4% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	documents.gnits.ac.in <1%
2	Publication	Nouf Abdullah Almujally, Bisma Riaz Chughtai, Naif Al Mudawi, Abdulwahab Alaz... <1%
3	Submitted works	Higher Education Commission Pakistan on 2010-11-22 <1%
4	Publication	Frank Y. Shih. "AI Deep Learning in Image Processing", CRC Press, 2025 <1%
5	Publication	C Cofaru. "Evaluation of digital image correlation techniques using realistic grou... <1%
6	Submitted works	CSU, San Jose State University on 2025-05-21 <1%
7	Internet	www.mdpi.com <1%
8	Internet	ijece.iaescore.com <1%
9	Internet	m.moam.info <1%
10	Internet	mro.massey.ac.nz <1%