Algerian People's Democratic Republic
Ministry of Higher Education and Scientific Research
Mohamed Boudiaf University - M'Sila

Faculty of Technology
Department of Electronics
N° :. . .. .

Field : Science and Technology
Department : Electronics
Specialization : STLC

Master's Thesis

To obtain the rank of

MASTER IN ELECTRONICS

Presented by
CHERGUI ASMA
ELGHARBI HADJIRA

Multiple CNN Models For Enhanced Palmprint Recognition

Composition of the jury :

Name & First Name	**Title & affiliate**	**Quality**
BRIK Youcef	Associate Professor, University-Msila	President
ATTALLAH Bilal	Associate Professor, University-Msila	Supervisor
NADIR Cheyma	PHD student, University-Msila	Thesis Co-Director
KHENNOUF Salah	Associate Professor, University-Msila	Examiner

Academic Year : 2023/2024

Acknowledgments

First and foremost, we are deeply grateful to Allah the Almighty for granting us the health, strength, and patience needed to complete this modest work. Without His blessings, nothing can be achieved or created.

We extend our heartfelt thanks to our parents for their unwavering moral support and encouragement.

Our sincere appreciation goes to our promoter, Dr. Bilal Attallah, and Dr.Cheyma Nadir for their invaluable guidance and advice.

We also wish to express our respect and gratitude to the members of the jury for their willingness to evaluate our work.

A special thank you is reserved for all the teachers in the Electronics Department at the University of M'Sila.

Dedication

I dedicate this Thesis

To my parents, who planted the seed of knowledge in my mind and nurtured it.

To my brothers and sisters.

To my husband, who has walked every step of this journey with me.

To all my family.

To all my friends.

To my partner Hadjer.

To my dear teachers and everyone who taught me a letter

To all those who have supported and encouraged me.

To all those who have a close or distant relationship with the accomplish this work.

Asma

Dedication

I dedicate this Thesis.

To my father and my mother.

In appreciation for the support, The sacrifices and all the effort they made for me. To my brothers.

To my sisters.

To my husband.

To all my family.

To all my friends.

To my partner Asma.

To all the professors and teachers.

I have had during All my school curriculum and which allowed me to succeed in my studies.

Hadjira

Abstract

These days, there is more talk about increasing crime, piracy, and lack of security across different sectors. It is also very important to verify people's identities for financial transactions, accessing services, and mobility. Traditional security systems use pre-existing information (like passwords or PINs) or token-based access (like keys, IDs, or badges). However, these systems frequently cannot discriminate between fraudsters and those who are allowed, they are less trustworthy in many environments.In this work, we choose to investigate one of these systems, which is a deep-learning palmprint recognition system. This system is difficult to replicate. There are several benefits, such as affordability and simplicity of usage. Our work may be categorized into two parts for feature extraction : transfer learning and fine-tuning, and two strategies : learning one instance and multiple instances. Firstly, we prepare our datasets into 3 datasets to evaluate our proposed models : left, right, and multiple instances. After that, we select three convolutional neural network algorithms to carry out the feature extraction and classification operation to confirm individual Recognition using both techniques : transfer learning and fine-tuning. The PolyU palmprint database is used to evaluate the performance of the suggested model. Our proposed method for the PolyU palmprint database using transfer learning achieved an accuracy of 85.25% with VGG16, 87% with DenseNet121, and 86.25% with MobileNetV2. Using fine-tuning, we achieved an accuracy of 92.50% with VGG16, 98.75% with DenseNet121, and 98.50% with MobileNetV2. Experimental results conclude that the proposed work obtained good performance compared to existing methods in multi-instance scenarios.

Keywords : Palmprint Recognition,CNN,Feature extraction,multi-instance , one-instance

Table List

Figure List

Abreviations list

AI Artificial intelligence
AWE Annotated Web Ears
BSIF Binary Statistical Image Feature
CNN Convolutional neural network
DCGAN Deep Convolutional Generative Adversarial Network
DL Deep learning
DNNs Deep Neural Networks
FN False Negative
FP False Positive
ILSVRC ImageNet Large Scale Visual Recognition Challenge
MAML Model-Agnostic Meta-Learning
ML Machine learning
RNN Recurrent Neural Networks
RGB Red Green Blue
TN True Negative
TP True Positives
UL Unsupervised learning
SL Supervised learning
RL Reinforcement learning
SSL Semi-supervised learning
VGG Visual Geometry Group
SVM using a support vector machine
HKPU Hong Kong Polytechnic University
IITD improved intrinsic time-scale decomposition
PCA Principal Component Analysis

Table of contents

Introduction general

These days, we often hear more about increased crime, piracy, and cross-sectoral instability. Furthermore, verifying people's identities is becoming more important due to the growth in the number and diversity of transactions (people's mobility, financial services transactions, access to services, etc.). Conventional security methods rely on either previously learned knowledge (passwords, PINs, etc.) or token-based items (keys, identifiers, badges, etc.). Because they frequently cannot discriminate between those who are genuinely authorized and those who are fraudulent, these systems are less reliable in many situations.

Several biometric techniques have been developed to control access to physical resources, enabling biometrics to easily adapt to numerous applications (such as airports, and other places) and logical resources (such as computers and bank accounts).

This thesis focuses on studying a deep learning palmprint recognition system, which uses a palmprint as a biometric. Palmprints were chosen for their unique identity, easy acquisition, high recognition accuracy, and distinctiveness.

The main objective of this work is to implement a comprehensive palmprint recognition authentication mechanism. The goal is to create a robust deep-learning biometric pattern extraction system that provides representations to reduce errors and enhance performance.

Two techniques are employed : transfer learning and fine-tuning. These techniques reduce training time and resource requirements by eliminating the need to train models from scratch. They improve performance with limited data, generalize across tasks, and provide accessible state-of-the-art performance without extensive expertise or computational resources.

Two learning strategies are used in this work : one-instance and multi-instance.One-instance learning uses a single biometric sample from the same subject, like one palmprint image per person. Multi-instance learning uses multiple samples of the same biometric trait, such as collecting palmprints from both hands. Both strategies are used in this work to see how they affect the performance and efficiency of the deep learning palmprint recognition system.

This thesis consists of three principal chapters :

1. Biometrics and Deep Learning : This chapter provides an overview of machine learning and deep learning, defines biometrics, and outlines the different system architectures.

2. The Proposed Palmprint Recognition System : This chapter presents the most popular approaches and strategies based on several databases for deep learning and machine learning-based Palmprint recognition and present our peoposed system.

3. Results and Discussions : In this chapter, the PolyU palmprint database is divided into two parts : training and testing. Two experiments are conducted : the first using transfer learning and the second using fine-tuning. Both experiments employ two learning strategies (one-instance and multi-instance) with three models (VGG16, DenseNet121, and MobileNetV2).

Chapter One

BIOMETRICS AND DEEP LEARNING

CHAPTER 1

BIOMETRICS AND DEEP LEARNING

1 Introduction

There are two sections in this chapter. The first portion of our chapter will be on biometrics. We will define them, describe the system's function, and review varieties. The second section will provide an overview of machine learning and deep learning.

2 Biometrics

Biometric information is linked to unique human traits. Biometric systems are among the most promising approaches to user authentication. Because biometrics make it difficult to impersonate or steal identities, they may be preferred over many conventional techniques like smart cards and passwords.

You can identify someone subject to access control and surveillance using the biometric recognition device. Generally speaking, physiological and behavioral traits fall under the category of biometric identifiers. Physical characteristics of the body, such as iris, fingerprints, palm veins, DNA, facial recognition, and so on, are referred to as physiological features[Thomas et al., 2016].

2.1 Biometric System

A biometric system generally consists of five interconnected components, as shown in Figure 1.1.

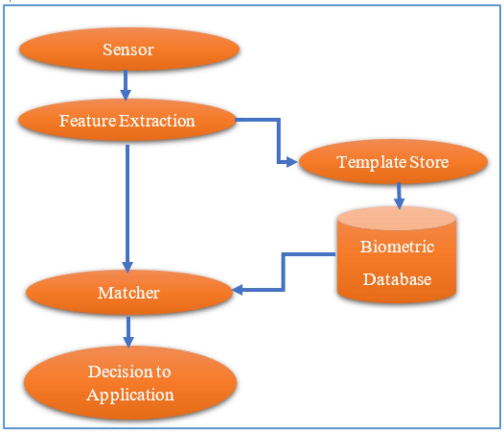

FIG 1.1 – *Generic Biometric Systems[Sabhanayagam et al., 2018]*

- Sensor module/ Image acquisition : to get an individual's raw biometric data, such as signals or audio, video, and picture data.

- Feature extraction Module : The process of utilizing computer vision, machine learning, and pattern recognition algorithms to extract distinct biometric information and produce templates.

- Database module : Users' biometric data was registered, and many user templates were saved. **- Matching module :** In order to detect similarities between two biometric samples, the presently retrieved characteristics are compared to stored templates to get a match score or value.

- Decision-making module : Comparing the findings to a specified limit determines whether the result is accepted or rejected

The following figure 1.2 represent the generic process(Enrollment, Verification and Identification) :

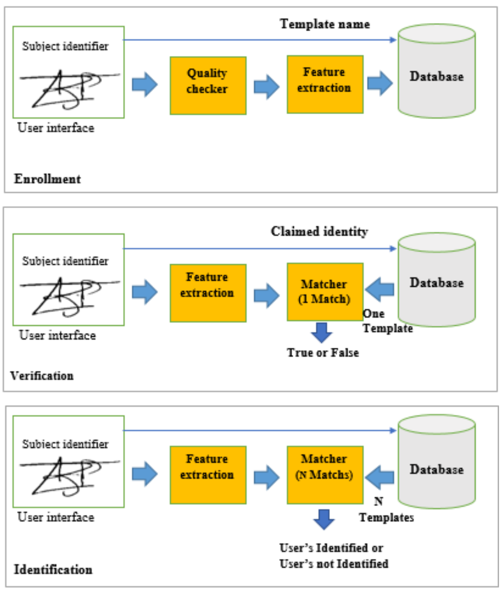

FIG 1.2 – *Enrollment,Verification and Identification[Dhanalakshmi and Srinivasan, 2013]*

A biometric system can be used for verification, authentication, or identification. Verification is a one-to-one procedure. It's sometimes referred to as authentication, and it's used to compare them to people who say they are who they say they are identification.

This is done to confirm the person's identification. Reject or accept the result in binary, based on the matching algorithm used. Acknowledgment is a one-to-many procedure. It is utilized to identify the biometric template with the highest similarity by comparing it to each registered template in the database.

This is employed to ascertain an individual's identification. Verification and identification are not the same thing. Recognition is the automatic process of comparing an individual's biometric data with all other individuals' biometric templates that are already in the database; this is known as a one-to-many match. [1 :M]. Verification is the process of confirming an individual's identification using a biometric that they have supplied. It is a one-to-one [1 :1] match between the biometrics that have been retrieved and recorded in the system. There are two ways that the biometric identification process functions : enrollment (registration) and authentication. The procedure of registering is capturing and preserving each person's distinctive qualities [Sabhanayagam et al., 2018].

2.2 Advantages and Disadvantages of Biometrics

The following table shows the Advantages and Disadvantages of biometrics.

Advantages	**Disadvantages**
Increase security	False positives
Can not be copied	Adaptability to rate of change
Can not be shared	scalability
Convenience	Difficult to control sensor
Auditable trial	complex Processor
Accuracy	Expensive
Can not be lost	Financial cost
Minimise paper work	Privacy issues

TABLE 1.1 – *Advantages and Disadvantages of Biometrics [Babich, 2012]*

2.3 Applications of Biometrics

Biometrics can be found in :

- For a variety of applications, including online banking, border control, e-commerce, medical records management, computer networks, verification, investigations, access control, and welfare disbursement, biometric systems are used to secure both the physical buildings and the surveillance systems
- To increase confidence following the confirmation that the person is authenticated and has all rights and privileges.

- For the dependable techniques' safe, secure, effective, and convenient operation of transactions.
- To recognize a specific person and manage who has access to data, locations, and services.
- Increasing security, lowering fraud, and enhancing safety.
- Offers stability, robust immunity, and distinctiveness against forgeries [Dargan and Kumar, 2020].

2.4 Biometric characteristics

Biometric modalities are the biometric traits that allow for the verification of an individual's identification. These modalities are predicated on the examination of personal data. The convergence of morphological or behavioral features for use in biometric applications is determined by a number of variables [Jain et al., 2007] :

- **University :** Each user with application access has to possess the quality.
- **Uniqueness(distinctiveness) :** The characteristic needs to differ enough across individuals.
- **Performance :** This function is fast and accurate in identifying people.
- **Permanence :** The biometrics of an individual must be sufficiently constant throughout time.
- **Measurability (Collectability) :** Biometric data should be able to be recorded and digitalized using the right tools.
- **Circumvention :** : In order to identify the features, the biometric system must be safeguarded or the measured characteristics must be impervious to counterfeit.
- **Acceptability :** To be utilized, the system has to fulfill a number of requirements (speed, convenience of use, etc.)

2.5 Biometrics modalities

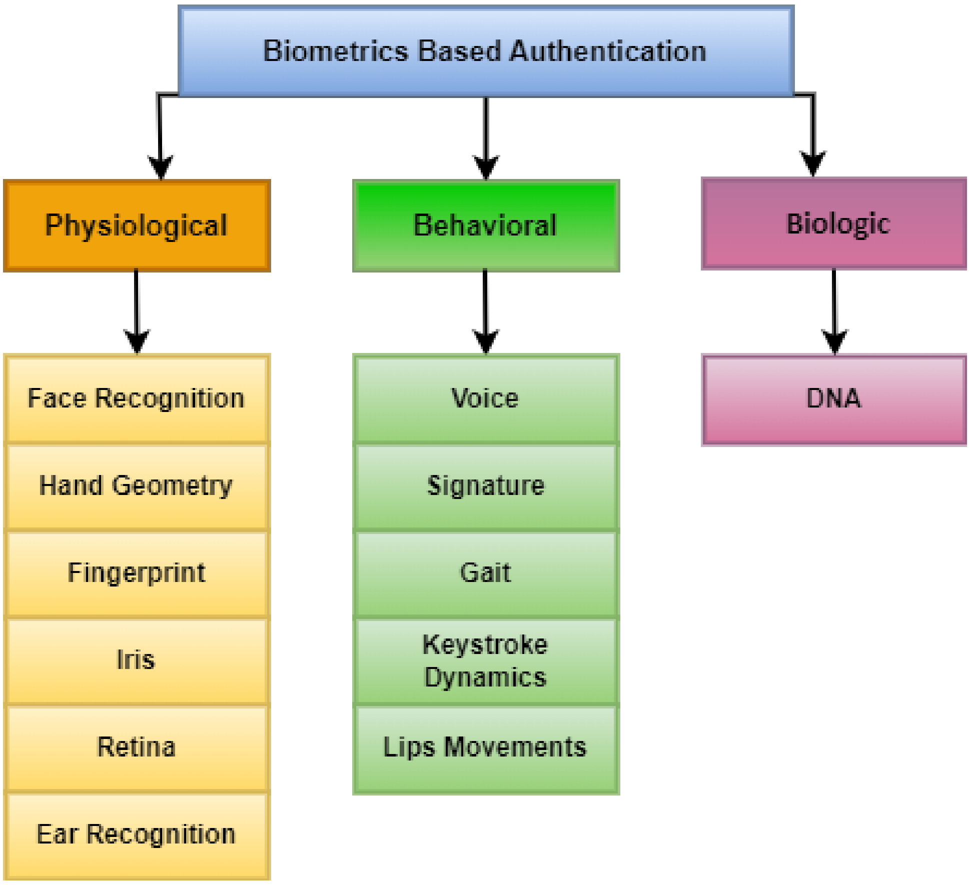

FIG 1.3 – *General classification of Biometric schemes*

2.5.1 Biometrics physiological

2.5.1.1 Ear : An additional biometric authentication method relies on identifying the distinct form and appearance of an individual's ear. People, of course, are born with a vision form of his or her ears. The human ear, however, does not alter as a person grows or even ages. Its consistent stability raises its security level as a suggested technique for people's security identification and verification[Alsaadi, 2015].

a) The Advantages :

- Static look and form.
- Faster identification
- Most stable and low computing complexity.
- Shorter processing times

b) The Disadvantages :

- Recognition error because the photos are not perfect.
- Difficulty identifying because of headgear, jewelry, and hair.
- • Not thought to be particularly unique.

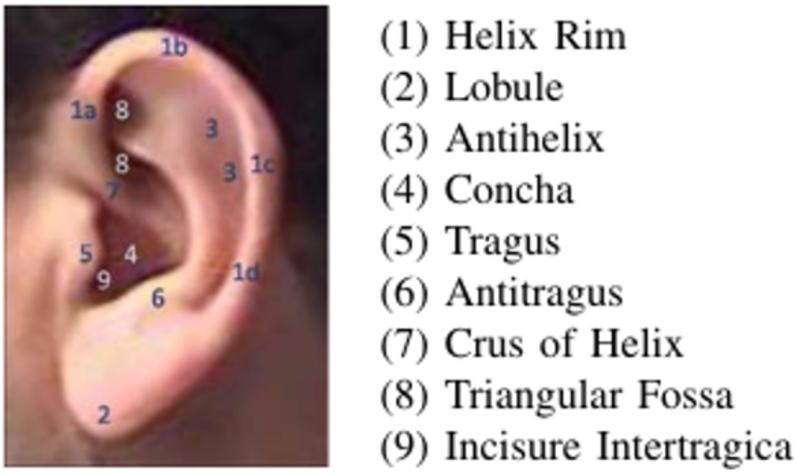

FIG 1.4 – *General classification of Biometric schemes[Sabhanayagam et al., 2018]*

2.5.1.2 Face : Every person has a distinct face, which may be used as a biometric profile for safe authentication. This is a universal reality. Face recognition systems have sprung from the concept of using faces for authentication. High-capacity cameras are employed to capture the face, which serves as a template for matching. Now, the template is compared using several pattern-matching methods in order to confirm or identify a certain identity[Harakannanavar et al., 2019].

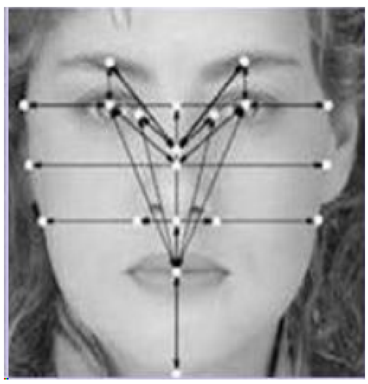

FIG 1.5 – *Face scanner and Examples of face database.[Sabhanayagam et al., 2018]*

a) The Advantages :

- Socially acceptable.
- Easy to save templates in database.
- Completely unobtrusive.
- requires no physical interaction
- Rapid processing of identification.

b) The Disadvantages :

- The face's expressions are erratic in terms of light and direction.
- There is no distinction between identical twins.
- The system can be tricked by physical changes.

2.5.1.3 Fingerprint Recognition : Fingerprints are graphic patterns of ridges and valleys on the fingertips. The termini and bifurcations of the ridge are referred to as minuae. Each person is uniquely identifiable from the others by their fingerprints. Invariance and singularity are the two fundamental assumptions that allow fingerprint identification. An invariant fingerprint is one whose characteristics don't change over time. The idea that no two persons have precisely the same fingerprint pattern is known as the singularity[Ali et al., 2016].

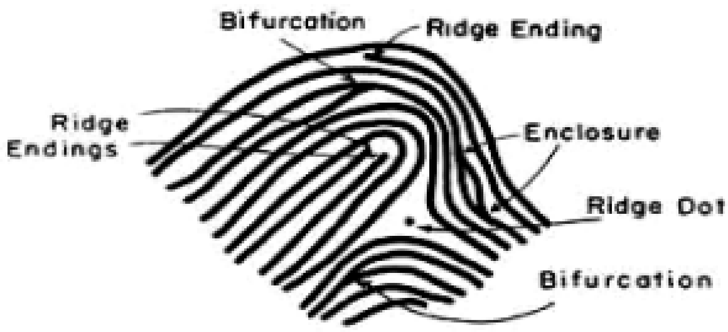

FIG 1.6 – *different ridge features on Fingerprint image[Ali et al., 2016]*

a) The Advantages :

- More dependable and safe.
- comparatively inexpensive.
- The template has a tiny size and matches quickly.

b) The Disadvantages :

- Artificial wax fingers are easily tricked.
- cuts, scars, or missing fingers can make identification more difficult.
- Some people's fingerprints were erased or degraded.

2.5.1.4 palmprint Recognition : Palmprint recognition is a biometric authentication method that uses the unique patterns of an individuals palm, including the lines, ridges, and minutiae, to identify and verify a persons identity. This technology captures and analyzes the palm's physical characteristics through imaging techniques and algorithms to match the collected data against a stored database of palmprints.[Amrouni et al., 2023].

a) The Advantages :

- More dependable and safe.
- comparatively inexpensive.
- High Accuracy and Reliability.

b) The Disadvantages :

- Limited Public Databases.
- cuts, scars, or missing palms can make identification more difficult.

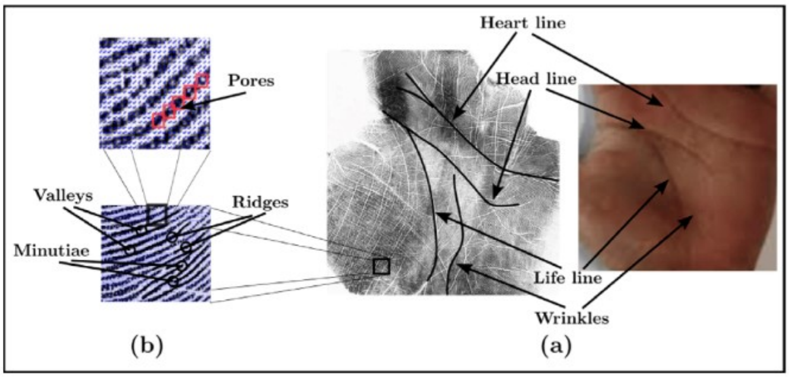

FIG 1.7 – *Palmprint*

2.5.2 Biometrics Behavioral

2.5.2.1 Signature Generally speaking, signature recognition technology is a behavioral biometric technique that identifies people by examining their written signatures, either online or offline . In the early decades, this technique was employed to differentiate between individuals . It is available for purchase both offline and online.Put another way, a signature can be formed by customary inc. and drawn on paper using a conventional method. Another option for obtaining a handwritten signature is through the use of technological devices, such smartphones. [Gandhe and Jawale, 2016],[Galbally et al., 2017],[Alsaadi, 2021].

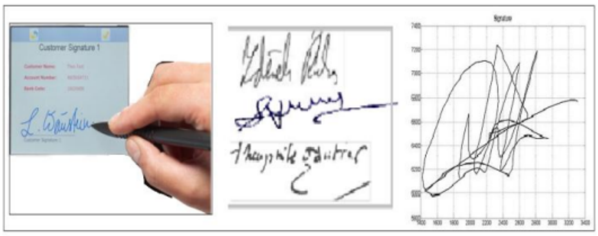

FIG 1.8 – *Signature samples[Sabhanayagam et al., 2018]*

a) The Advantages :

- Is widely accepted by the people.
- Accuracy that is reasonable.
- If the template is taken, recovery is simple.

b) The Disadvantages :

- Individual signatures can vary over time, and even the same person's signatures might be inconsistent.
- Professionals manipulate the system.
- users ought to be aware of this while using signature tables.

2.5.2.2 Keystroke : The way that a person types on a keyboard falls into the topic of behavioral traits of people. The technique looks at important stroke characteristics, such as time. the time, pressure, and pace at which a person types the password[Harakannanavar et al., 2019].

FIG 1.9 – *Keystroke[Harakannanavar et al., 2019]*

a) The Advantages :

- Knowledge can be identified quickly and safely ; it doesn't require specialized equipment or brand-new, innovative sensors ; and it is reasonably priced.
- It's not necessary for writers to be terrified of being noticed.
- People who write need not be afraid to be observed.

b) The Disadvantages :

- Harder to use comfortably.
- modify the input rhythm due to illness, a new keyboard, etc.

2.5.2.3 Voice : : Utilizing factors such as pitch period, vocal tract, mouth, nasal, and lip movements, voice recognition systems may recognize individuals. Because aging, illness, and emotional situations may all affect speech, these methods are frequently employed. Users may identify different utterances with text-independent systems, but text-dependent systems, which are based on Hidden Markov Models, are more common[Kaur et al., 2014].

FIG 1.10 – *Voice Biometrics[Sabhanayagam et al., 2018]*

a) The Advantages :

- Simple to use
- no new equipment is needed.

b) The Disadvantages :

- The rate of false mismatches is large
- it is readily falsified.

2.5.3 Biometrics biologic

2.5.3.1 DNA Currently, DNA identification requires an invasive approach and requires samples of tissue, blood, saliva, semen, hair, etc. for the authentication procedure[Harakannanavar et al., 2019].

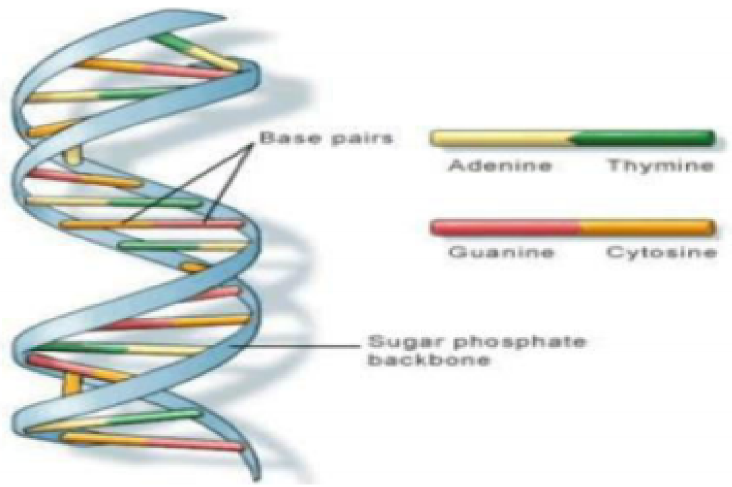

FIG 1.11 – *DNA structure[Harakannanavar et al., 2019]*

a) The Advantages :

- Provides the greatest accuracy.
- Allows for two individuals to have a DNA profile that is less than one percent billion same.

b) The Disadvantages :

- Prolonged sample collecting times necessary to produce the desired outcomes.
- Despite the abundance of information, privacy issues still exist.
- The disintegration of the sample has an impact on the results.

2.6 Classification of Biometric Systems

2.6.1 Unimodal Biometric System

A system that relies only on one biometric characteristic or information source for authentication or verification is called a unimodal (or single) biometric system. Although unimodal systems are believed to have progressively increased in accuracy and reliability, as previously stated, they frequently have issues during the enrollment process due to non-universal biometric traits, spoofing, and inaccurate data from noisy sources. Moreover, in practical applications, unimodal biometric systems often perform less well than intended. Hence, using a multimodal biometric authentication system is one way to address these issues. There is further discussion of the problems surrounding unimodal biometric systems.[Oloyede and Hancke, 2016].

2.6.1.1 Noisy Data : In biometric data, noise is typically inherent to poorly maintained sensors. The presence of dirt on a fingerprint scanner's sensor, which results in a loud fingerprint, is a common occurrence. Not being able to speak in the right voice noisy data also occurs during enrollment. Additionally, blurry photographs of the face and iris might result from improper camera focusing[El-Abed et al., 2010],[Mittal and Garg, 2014].

2.6.1.2 Non-Univarsality : If every user can utilize a biometric attribute to identify themselves, then a biometric system is considered universal. But not every biometric feature is present everywhere. It has been disclosed that approximately A small percentage of the population—2%, specifically—are unlikely to have a high-quality fingerprint; these people may have impairments or experience other difficulties registering, which makes it impossible for them to be properly added to the system's database[El-Abed et al., 2010],[Mittal and Garg, 2014].

2.6.1.3 Lack Of Individuality : A biometric system's traits, like those of a face recognition system that works with facial photographs, may be somewhat comparable. Father and son, or identical twins, are common examples. This results in a rise in the false match rate, which is caused by the uniqueness problem [El-Abed et al., 2010],[Mittal and Garg, 2014].

2.6.1.4 Susceptibility To Circumvention : It is feasible for an impostor to mimic an individual's characteristics by creating fake characteristics. For example, creating false fingers using fingerprints and utilizing them to incorrectly access a biometric system[Cao et al., 2012].

	Biometric Sensing Systems									
Factors	Finger print	Face	Hand Geometry	Iris	Voice	Hand Signatory	Gait	Ear	Palm Vein	Palmprint
Accuracy	High	Low	Medium	High	Medium	Medium	High	High	High	High
Ease of Use	High	Medium	High	Medium	High	High	Medium	Medium	Medium	Medium
Cost	Low	Medium	Medium	High	High	Low	High	High	High	High
Privacy	High	High	Medium	High	High	High	High	Low	Low	Medium
Distinctiveness	High	Low	Medium	High	Low	Medium	Medium	High	High	High
Error Causing Factor	Age	Occlusion	Injury	Eye Angle	Illness	Inconsistency	Weight Gain	Pose	Illness	Age
Barrier to Universality	Worn Ridges	Plastic Surgery	Hand Impairment	Visual Impairment	Speech Impairment	Forging	Drunken-ness	Lighting Conditions	Ageing	Worn prints

TABLE 1.2 – *Performance of the various biometric sensing systems[Oloyede and Hancke, 2016]*

2.6.2 Multimodal Biometric System

Multimodal biometric systems are more reliable than unimodal systems since they combine data from multiple biometric features for individual identification. Due to non-universality issues, unimodal systems might not be suitable for all applications. In contrast, multimodal systems can offer excellent accuracy and security. Several technologies can be effectively integrated into multimodal biometric systems, reducing vulnerability to spoofing and increasing overall system performance. Each of the four typical biometric system modules—sensor, feature extraction, matching, and decision-making—has its own advantages. This multimodal approach increases overall system robustness and lowers failure to enroll (FTE) rates.[Dahea and Fadewar, 2018]

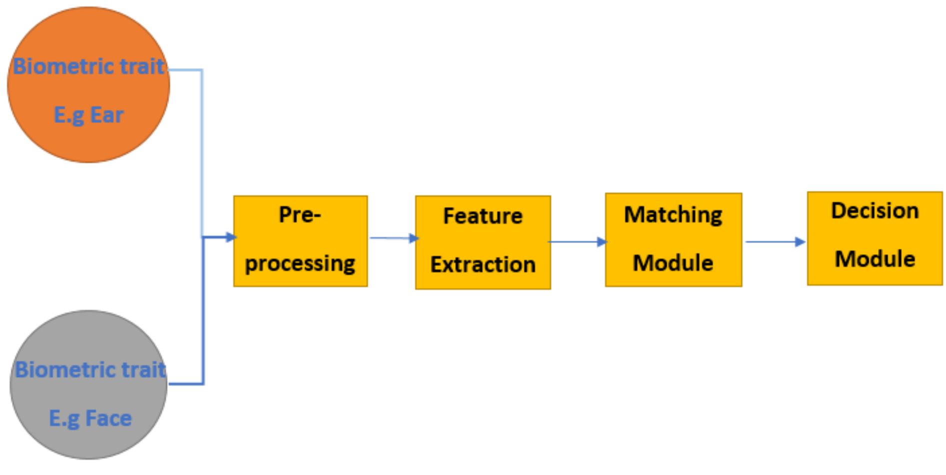

FIG 1.12 – *Process Involved in a Multimodal Biometric System[Ross and Jain, 2004]*

Present relevant research in the area of multimodal biometric systems.

Researcher[s]	Year	Multibiometric Properties
Antonio iula.[Vyas et al., 2022]	2022	3D hand geometry and 3D palmprin
NabilHezil. [Hezil and Boukrouche, 2017]	2017	human ear and palmprint
MitulDdhameliya [Dhameliya and Chaudhari, 2013]	2013	Palmprint and fingerprint
BasmaAmmour. [Ammour et al., 2017]	2017	face and iris

TABLE 1.3 – *Demonstrate related work in the field of multimodal biometric systems.*

In2022, A[Vyas et al., 2022] conducted his study , a multimodal ultrasonic identification system combining 3D palmprint characteristics and hand geometry is proposed. The method creates a 3D template by extracting and combining 2D photos. The outcomes of the experiments demonstrate that fusion greatly enhances recognition performance, resulting in an Equal Error Rate of 0.08% and a 100% identification rate.

In2017, A[Hezil and Boukrouche, 2017]conducted his study, In order to provide reliable human identification, the study investigates the feature-level integration of ear and palmprint biometric modalities utilizing local texture descriptors. The suggested multimodal biometric approach can boost recognition rates by 100%, according to experiments conducted on the IIT Delhi-2 palmprint and ear databases.

In2017, A[Dhameliya and Chaudhari, 2013]conducted his study The suggested multimodal biometric system compares data templates using the Euclidean-distance matching method and combines processed information using a fusion mechanism. It attains ideal False Acceptance and False Rejection Rates with an accuracy of 87% in recognition.

In2013, A[Ammour et al., 2017] conducted his study, With the use of fusion rules and normalization techniques, the multi modal biometric system-which uses the face and iris for identification-was verified on huge datasets and demonstrated high recognition rates.

2.7 Types of multi-biometric systems

Research on multi-biometrics is an intriguing and stimulating field. It is employed to identify people in order to raise security standards. Current research indicates that real-time applications of the next generation of biometrics are in the works.

Utilize these techniques : Multiple sensors, multiple algorithms, multiple instances, multiple samples, and multiple moda[El-Sayed, 2015] As shown in the figure 1.12 :

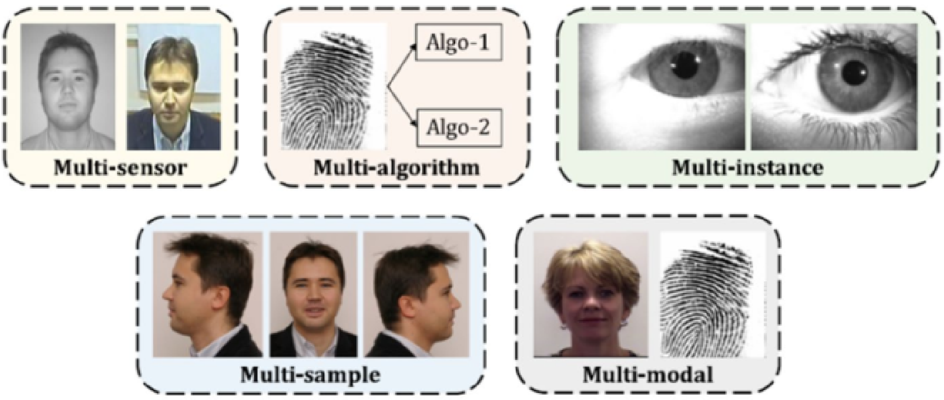

FIG 1.13 – *Types of multi-biometric systems [El-Abed et al., 2010]*

2.7.1 Multi-sensor

to record the same biometric, utilize many sensors. The face recognition system, for instance, can use an infrared camera and a visible light camera to capture the subject's face and gather other data that improves precision

2.7.2 Multi-algorithm

Optimize matching performance by integrating several feature extraction methods on a single biometric characteristic. The cost of these systems is reduced since they don't need more sensors, however integrating several algorithms will make computing more difficult.

2.7.3 Multi-instance

utilize the same biometrics more than once. For instance, all of the subject's fingerprints from both hands are gathered. If the system gathers instances sequentially from a single sensor, it might raise the time spent processing.

2.7.4 Multi-sample

capture the same biometric with a few modifications using a single sensor. To increase system dependability, the facial recognition system, for instance, can record the subject's face from several perspectives.

2.7.5 Multimodal

get many biometrics simultaneously. In addition to being more dependable and performing better in terms of withstanding spoofing, such a system often has a greater computing cost.

3 Deep learning and Machine Learning

Artificial intelligence(AI) is the engineering and science of creating intelligent devices, particularly computer programs. While the aim of utilizing computers to comprehend human intellect is comparable, artificial intelligence (AI) is not limited to techniques that may be seen through biological means[McCarthy et al., 2007]

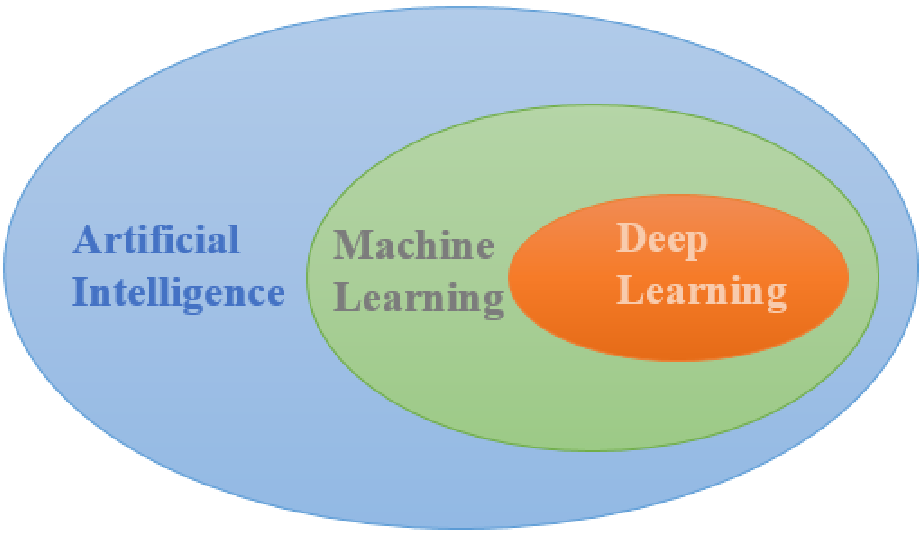

FIG 1.14 – *Deep learning Overview*

3.1 Machine Learning

Machine learning is a technique that teaches machines to efficiently handle data, especially when interpretation is difficult. With the abundance of available datasets, the demand for ML is increasing, and various approaches are used to make machines learn independently, especially with large data sets[Mahesh, 2020].

3.1.1 Types of Machine Learning Algorithms

The most used algorithms :

3.1.1.1 Unsupervised learning (UL) : These are referred to as unsupervised learning because, in contrast to the supervised learning described above, there is no teacher and no right or wrong responses. It is up to the algorithms to find and display the intriguing structure within the data. Few characteristics are learned from the data by the unsupervised learning algorithms. It recognizes the class of the data when it is introduced by using the previously learnt characteristics.Its primary applications are in feature reduction and clustering [Mahesh, 2020].

3.1.1.2 Supervised learning (SL) : is frequently used in classification issues as the objective is frequently to teach the computer to recognize a custom categorization system. Once more, one typical application of categorization learning is the recognition of digits. Generally speaking, categorization Any situation where a categorization may be inferred and shown to be beneficial is one for which learning is acceptable. If the agent is capable of classifying problems on its own, it may not even be required to assign predefined categories to each instance of a problem. This would be an illustration of unsupervised learning in the context of categorization [Ayodele, 2010]

3.1.1.3 Reinforcement learning(RL) is a branch of machine learning that studies how software agents should behave in a given situation to optimize a concept of total benefit. Along with supervised learning and unsupervised learning, reinforcement learning is one of the three fundamental paradigms in machine learning [Mahesh, 2020].

3.1.1.4 Semi-supervised learning(SSL) is a machine learning technique that combines supervised and unsupervised approaches. It can be beneficial in such machine learning domains and It takes a lot of work to obtain the labeled data by data mining in situations when the unlabeled data is already there. More popular supervised machine learning techniques involve using a ”labeled” dataset, where each record contains the result information, to train a machine learning algorithm. Below is a discussion of a few Semi Supervise learning methods [Mahesh, 2020].

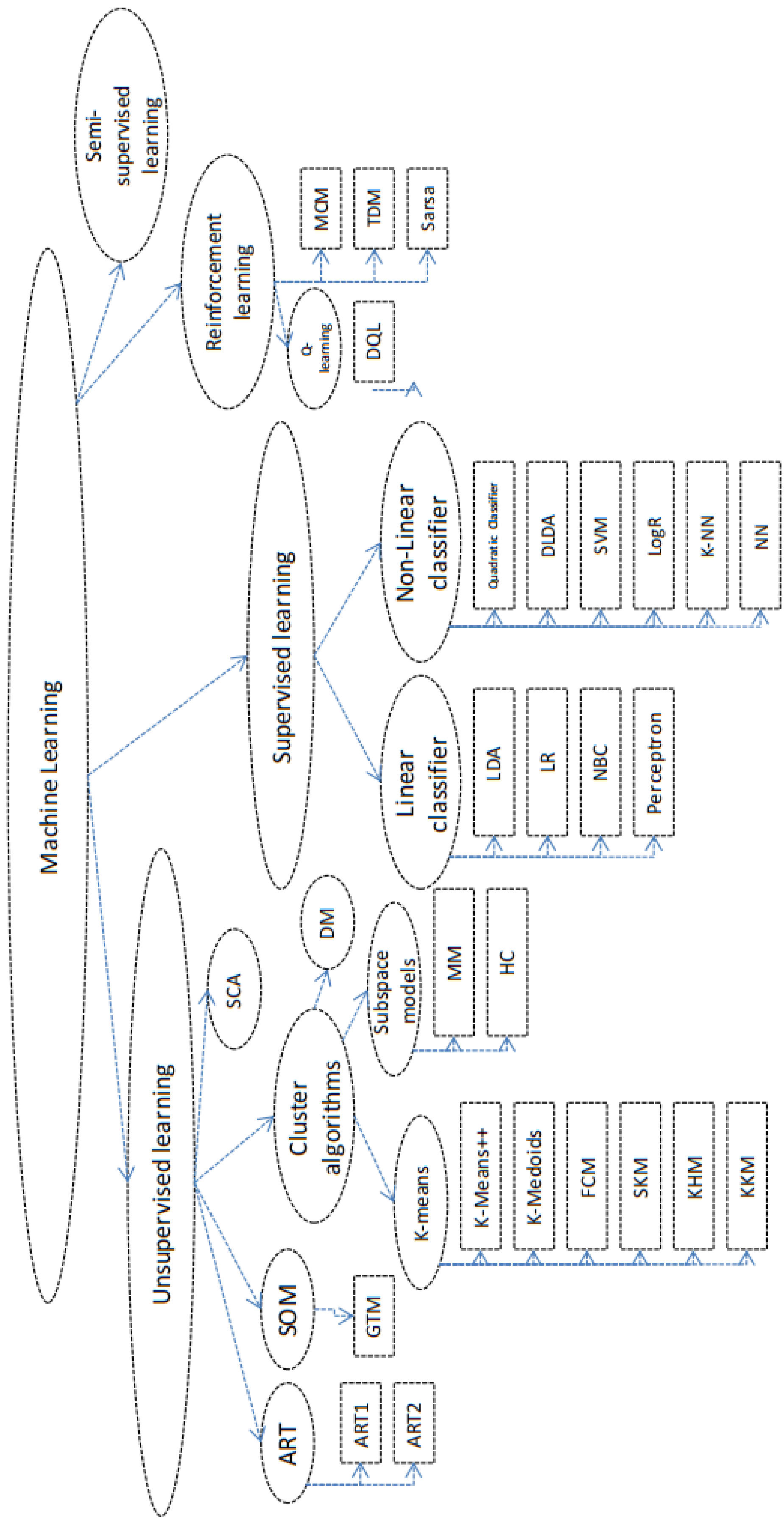

FIG 1.15 – *Taxonomy of ML algorithms*[Tiwari et al., 2015]

3.2 Deep learning

Deep learning, a subset of machine learning involving multiple processing layers, has significantly advanced various fields including speech recognition, visual object recognition, object detection, drug discovery, and genomics. This technique employs the back-propagation algorithm to uncover complex patterns in large datasets. Furthermore, deep convolutional neural networks and recurrent neural networks have revolutionized the processing of sequential data [LeCun et al., 2015].

3.2.1 Deep Autoencoder

These networks change input patterns into a compressed hidden representation by using an inherent statistical structure[Kriegeskorte and Golan, 2019]

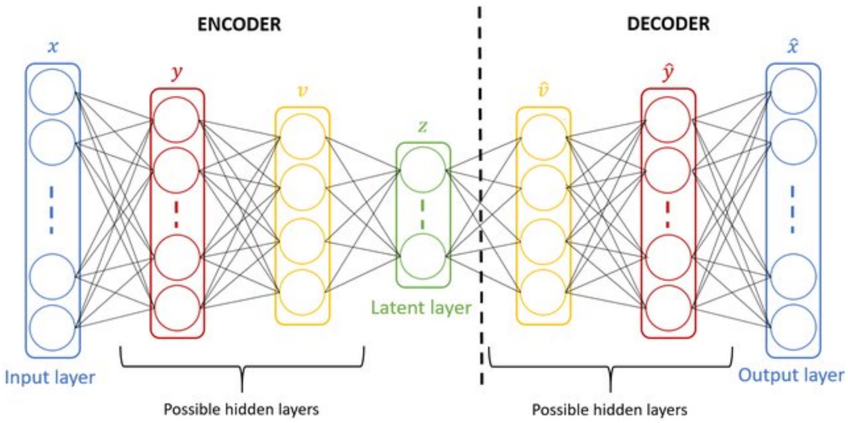

FIG 1.16 – *the architecture of the deep autoencoder model[Ross and Jain, 2004]*

3.2.2 Deep Recurrent Neural Network

is a type of artificial neural network that adds loops to the connections of a standard feedforward neural network. Unlike a neural network that is feedforward, An RNN's recurrent hidden state, whose activation at each step depends on the preceding phase's activation, allows it to handle sequential inputs. The network can display dynamic temporal behavior in this way[Mou et al., 2017].

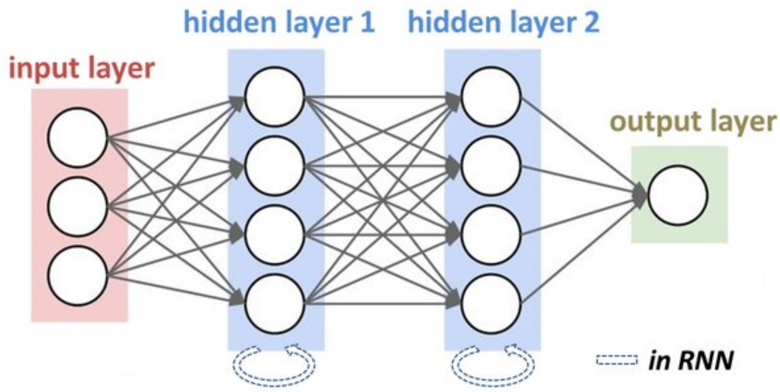

FIG 1.17 – *the architecture of the Deep Recurrent Neural Network[Murphy, 2016]*

3.2.3 Deep Convolutional Neural Network

Deep Convolutional Neural Networks (CNN) are powerful learning algorithms used in various applications such as image classification, object detection, video processing, natural language processing, and speech recognition. With the availability of large amounts of data and advancements in hardware technology, researchers have explored various ideas to improve CNNs. Recent innovations in CNN architectures have been categorized into seven categories based on spatial exploitation, depth, multi-path, width, feature-map exploitation, channel boosting, and attention. This survey provides an elementary understanding of CNN components, current challenges, and applications[Khan et al., 2020].

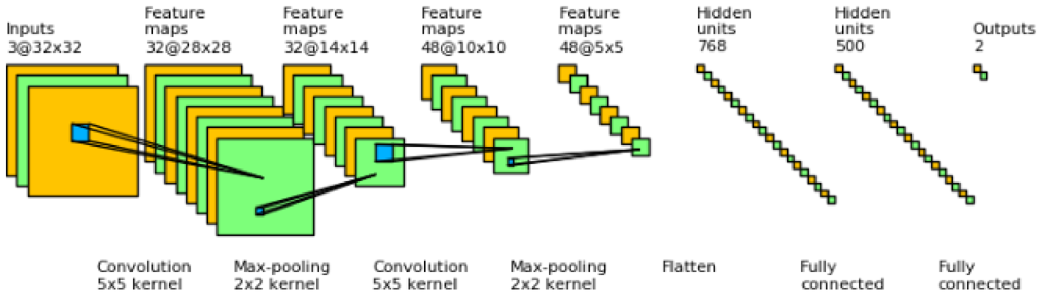

FIG 1.18 – *the architecture of the Deep Convolutional Neural Network*

A CNN model is composed architecturally of numerous layers that translate input images into output labels. The functions of these layers allow them to be separated into input, pooling, convolutional, ReLU, fully connected, and output layer, as displayed below :

3.2.3.1 Input Layer Neural networks receive input from images and represent it as a stack of matrices (like higher-order tensors) in the input layer[Zhang et al., 2019].

3.2.3.2 Convolutional layer : A ConvNet is a network that uses a convolutional neural network (CNN) to compute features. It consists of a layer of feature maps and neurons, with learnable filters or kernels convolved with the feature maps to produce a 2-dimensional activation map. The receptive field is a hyperparameter that controls the output volume. ConvNets are trained using backpropagation and backward pass convolution operations. A variant of traditional CNN is ”Network In Network” (NIN), which uses a Multi-Layer Perceptron (mlp) convolution filter and a global average pooling layer. NIN enhances the abstraction ability of latent concepts and can perform object recognition. Recently, CNNs have been adapted for dense prediction problems like semantic segmentation, using dilated convolutions and a rectangular prism of convolution layers[Aloysius and Geetha, 2017].

3.2.3.3 Pooling layers : Basic ConvNet architecture have alternating conv layers and pooling layers and the latter acts to lower the spatial dimension of the activation maps (without loss of information) and the number of parameters in the net and thus decreasing the overall computational complexity. This manages the overfitting issue. Typical pooling procedures include max pooling and average pooling [Aloysius and Geetha, 2017].

3.2.3.4 ReLU Layer : A small adjustment to ReLU produced Leaky ReLU, which is represented by Equation[Zhang et al., 2019]

$$\mathbf{f(x)} = \begin{cases} 0, & \text{if } x < 0, \\ x, & \text{otherwise.} \end{cases}$$

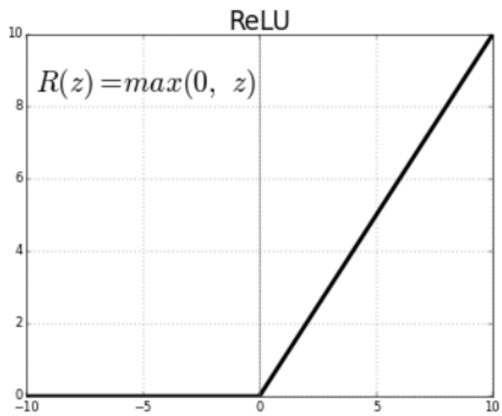

FIG 1.19 – *function of Rectified Linear Units [Sultan et al., 2019]*

Leaky ReLUs produce a negative value that scales with the input but at a slope significantly less than 1, as opposed to staying at zero for all input values less than zero. The resulting piecewise-defined function resembles the ReLU, but it has a small counterclockwise rotation of the zero portion into Quadrant III. ReLUs may occasionally become ”dead” if they become trapped in the area of the zero output plateau. Conversely, leaky ReLUs don’t have this propensity to ”die” during training[Murphy, 2016].

3.2.3.5 Fully Connected Layer : As in a typical neural network, every neuron in this layer is completely linked to every other neuron in the layer before it. Elevated This is where logic is used. There cannot be a convolution layer following a completely linked layer since the neurons are not spatially oriented (one dimensional). Recently, the fully linked layer in certain architectures has been replaced with a global average pooling layer, as in Lin et al.'s "Network In Network" (NIN)[Aloysius and Geetha, 2017].

3.2.3.6 Output Layer : The output layer yields the classification results[Mahesh, 2020].

4 Conclusion

We have covered biometrics and an overview of deep learning and machine learning in this chapter. We will discuss deep transfer learning for palmprint identification in the upcoming chapter, along with the methods that each researcher has employed to make it more effective.

Chapter Two

THE PROPOSED PALMPRINT RECOGNITION SYSTEM

CHAPTER 2

THE PROPOSED PALMPRINT RECOGNITION SYSTEM

1 Introduction

Over the years, considerable advancements have been made in palmprint identification. In this chapter, we outline the state-of-the-art in palmprint recognition technology to address these advancements and how deep learning and machine learning employ it. Next, we present our system and its constituent parts for palmprint recognition. To provide an overview of the methodologies and approaches employed in this work, a full description of every component of our system is provided. Lastly, we discuss the distinctions between the architectures employed in this work and the rationale for our system's use of deep learning.

2 PALMPRINT RECOGNITION STATE-OF-THE-ART

2.1 Using Machine learning

The following figure2.1 represents a classification method in machine learning

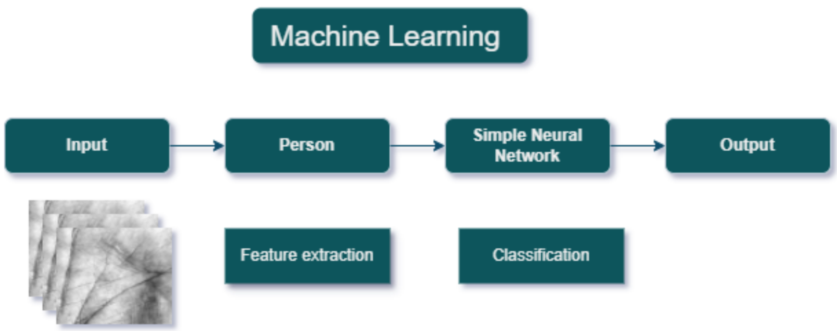

FIG 2.1 – *Palmprint recognition using machine learning*

Survey of method Machine learning

This table shows comparisons of different machine learning methods to identify palmprint recognition. With a variety of data from previous years.

Reference	Year	Method	Dataset	Accuracy(%)
[Tamrakar and Khanna, 2016]	2016	BGDPPH+KDA	CASIA	99.22
[Gumaei et al., 2018]	2018	HOG-SGF	CASIA	97.75
[Rida et al., 2018]	2018	SR	Polyu-MS	99.24
[El-Tarhouni et al., 2019]	2019	PCMLBP+PHOG	POLYU	99.34
[Amrouni et al., 2022]	2022	BSIF+DWT	IITD	98.77
[Zhao et al., 2022]	2022	DC-MDPR	IITD	98.43
[Wan et al., 2023]	2023	LR-2DLDGE	Polyu	93.87

TABLE 2.1 – *Survey of method Machine learning*

2.1.1 Machine Learning

The biometrics feature extraction and matching stage has made use of a few machine learning algorithms. These kinds of methods have also shown to be effective in matching, feature extraction, and improving the palmprint recognition (PPR) method's performance. The matching stage of PPR is when machine learning classifier-based approaches are used in the majority of PPR procedures. On the other hand, traditional palmprint approaches use distance-based techniques in the matching phase. Nearly all of the suggested machine learning palmprint algorithms have an accuracy rate of approximately 100%.

in 2016[Tamrakar and Khanna, 2016],The work provides a strong BGDPPH descriptor for palmprint ROI estimation, lowers noise and computational overhead, and improves identification rate by weighted score level fusion and dimension reduction.

- used method BGDPPH+KDA and database CASIA,reached accuracy equal to 99.22%.

in 2018[Gumaei et al., 2018],This paper presents a hybrid feature extraction method, HOG-SGF, that combines the histogram of oriented gradients (HOG) with a steerable Gaussian filter for palmprint recognition. Experimental results show the approach outperforms existing methods, even with limited training samples.

- used method HOG-SGF and database CASIA,reached accuracy equal to 97.75%.

in 2018 [Rida et al., 2018],The paper presents a sparse representation for classification method for palmprint identification, reducing sensitivity due to limited training data and showing promising results on multispectral and PolyU palmprint datasets.

- used method SR and database Polyu-MS,reached accuracy equal to 99.24%.

in 2019[El-Tarhouni et al., 2019],Using a multiscale local binary pattern, the study suggests an ensemble learning paradigm that addresses over fitting in linear discriminant analysis. Additionally, it presents a novel feature extraction method that outperforms the state-of-the-art methods.

- used method PCMLBP+PHOG and database Polyu,reached accuracy equal to 99.34%.

in 2022[Amrouni et al., 2022],The three-step method for palmprint recognition presented in this paper focuses on feature extraction and classification. Pre-processing, multiresolution analysis, and K-NN-based classifiers are used in this method.

- used method BSIF+DWT and database IITD,reached accuracy equal to 98.77%.

in 2022 [Zhao et al., 2022], The paper introduces a double-cohesion learning-based palmprint recognition method, enhancing recognition performance by leveraging palmprint features from multiple views, reducing computational complexity, and achieving superior results.

- used method DC-MDPR and database IITD, reached an accuracy equal to 98.43%.

in 2023[Wan et al., 2023],The 2D local preserving projections algorithm is enhanced with low-rank local discriminant graph embedding, ensuring local neighborhood discrimination, independent data points, and reduced noise and corruption.

- used method LR-2DLDGE and database POLYU,reached accuracy equal to 93.87%.

2.2 Using Deep learning

The following figure represents a classification method in Deep learning

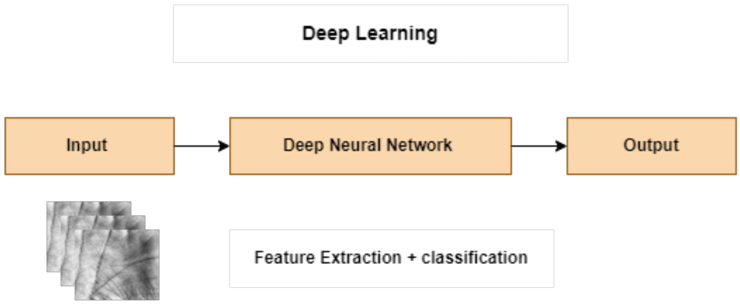

FIG 2.2 – *Palmprint recognition with using deep learning*

Survey of method deep learning

This table shows comparisons of different Deep learning methods to identify palmprint recognition. With a variety of data from previous years.

Reference	Year	Method	Dataset	Accuracy(%)
[Izadpanahkakhk et al., 2018]	2018	Fast CNN	IITD	94.70
[Matkowski et al., 2019]	2019	EE-PRnet	CASIA	97. 65
[Chai et al., 2019]	2019	PalmNet+GenderNet	IITD	99.01
[Genovese et al., 2019]	2019	PalmNet+GaborPCA	REEST	97.16
[Liu and Kumar, 2020]	2020	RFN	IITD	99.20
[Shen et al., 2022]	2022	ArcPalm-IR50+PTD	CASIA	99.85
[Türk et al., 2023]	2023	CNN deep features+SVM	Polyu	99.72

TABLE 2.2 – *Survey of method deep learning*

2.2.1 deep learning

in 2018[Izadpanahkakhk et al., 2018],A new open-source palmprint verification system employs convolutional neural networks and transfer learning to extract deep ROIs from palmprints, achieving high accuracy and efficiency.

— used method Fast CNN and database IITD,reached accuracy equal to 94.70%.

in 2019[Matkowski et al., 2019],A deep learning algorithm was used to create the new palmprint database NTU-PI-v1, which performs better than current techniques in identifying palmprints in contactless contexts, including images of children pornography and terrorists.

— used method EE-PRnet and database CASIA,reached accuracy equal to 97.65%.

in 2019[Chai et al., 2019],The paper presents a CNN model for palmprint identification and gender classification, utilizing two palmprint datasets, BJTU-PalmV1 and BJTU-PalmV2, demonstrating high accuracy and strong generalization ability.

— used method PalmNet+GenderNet and database ITTD,reached accuracy equal to 99.01%.

in 2019[Genovese et al., 2019],Current methods using CNNs and local texture descriptors are surpassed by touchless palmprint recognition systems like PalmNet, which use Gabor responses and Principal Component Analysis to improve recognition accuracy.

used method PalmNet+GaborPCA and database REEST,reached accuracy equal to 97.16%.

in 2020[Liu and Kumar, 2020],The study offers a deep learning framework with a faster-R-CNN architecture, soft-shifted triplet loss function, and fully convolutional network for contactless palmprint detection that outperforms existing techniques.

- used method RFN and database ITTD,reached accuracy equal to 99.20%.

in 2022[Shen et al., 2022],The study presents a cross-device palmprint identification benchmark using a distribution-based loss function, surpassing SOTA rivals and other biometric techniques across a range of benchmarks.

used method ArcPalm-IR50+PTD and database CASIA,reached accuracy equal to 99.85%.

in 2023[Türk et al., 2023],The study presents a hybrid approach combining deep learning and machine learning for palmprint recognition, achieving an overall accuracy of 99.72% and a low execution time of 0.10 seconds.

used method CNN deep features+SVM and database POLYU,reached accuracy equal to 99.72%.

3 PROPOSED PALMPRINT RECOGNITION SYSTEM

This section presents the proposed method in detail. We begin by describing the strategies of learning single-instance and multiple-instance recognition after that we describe the palmprint feature extraction stage. At this stage, we used three CNN models based on transfer learning and fine-tuning for deep feature extraction. Finally, a classification step is performed using Softmax for multiclass classification in our task. The proposed palmprint recognition architecture is illustrated in fig2.3.

3.1 Learning strategies

In biometric systems, differentiating between single and multiple instance data collecting is essential. A single instance is defined as using one biometric sample from the same subject, such as a single palmprint image per individual. In contrast, a multiple

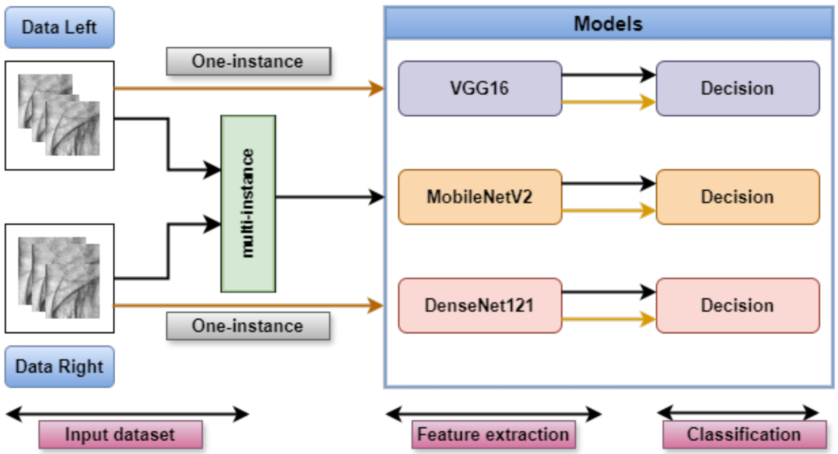

FIG 2.3 – *The proposed system framework*

instance strategy requires collecting numerous samples of the same biometric feature from the same person, such as palmprints from both hands. For example, in a single instance scenario, in our study we have six images per person from 100 individuals, we might have 400 images from the left hand 400 from the right hand for training, and 200 images from each hand for testing. This totals 600 images per hand, combining training and testing data. In contrast, in multiple instances, we have 800 images (400 left hand and 400 right hand) for training and 400 images (200 left hand and 200 right hand) for testing, for a total of 1200. Thus, each individual provides many features from both hand of the same person, resulting a more robust biometric system .

3.2 Feature extraction using deep learning

Feature extraction maximizes differences and similarity among people, with algorithms developing rapidly in recent decade. They focus on accurate and effective feature recording [Zhong et al., 2019]. In this study, we utilized CNN architecture for palmprintrecognition. To avoid issues associated with finding the optimal parameters and achieving the best accuracy when working with CNNs, we considered and tested four descriptor-based deep CNN models : VGG 16, MobileNet V2 and DenseNet-121.

3.2.1 VGG16 Model

VGG 16 is a 16-layer CNN model that is considered one of the best and most effective models today. It focuses on ConvNet layers with a 3x3 kernel size and is free to download for use in systems and applications. The model's minimum expected input image size is 224 x 224 pixels with three channels. Optimization algorithms evaluate whether a neuron must be engaged by determining the weighted sum of input. The kernel function is needed to induce non-linearity into the output neuron. The input layer and activation function add non-linearity to artificial neural networks. The model's values are freely available online for use in systems and applications[Younis et al., 2022].

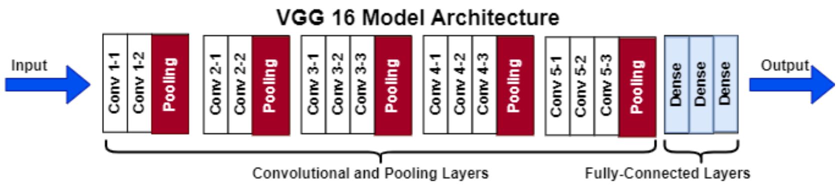

FIG 2.4 – *The architecture VGG16 Model*

3.2.2 MobileNetV2

MobileNet V2 is improves mobile model performance by replacing expensive convolutional layers with depthwise separable convolutional blocks. Each block consists of a 3x3 depthwise convolutional layer and a 1x1 pointwise convolutional layer, resulting in faster results. This architecture is faster than regular convolution and has a bottleneck residual block, reducing the number of channels in the output[Michele et al., 2019].

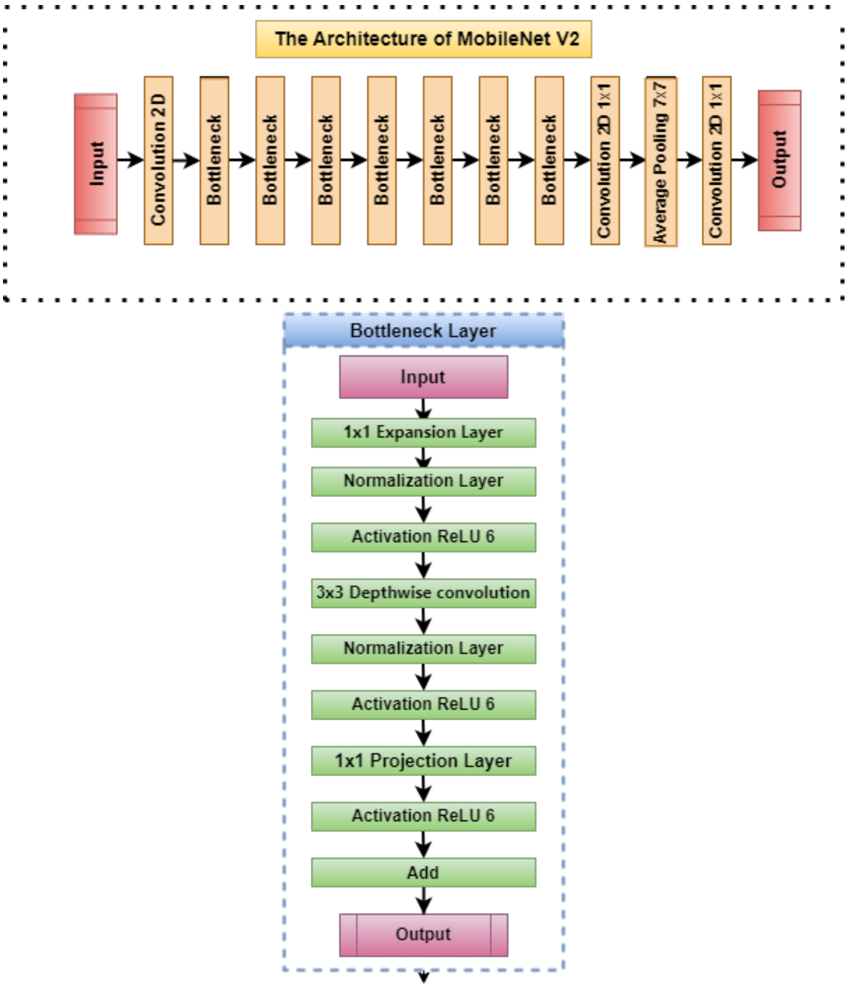

FIG 2.5 – *The architecture MobileNet V2 Model*

3.2.3 DenseNet 121 Model

Optimized DenseNet is a convolutional neural network with dense connections, aiming for maximum information transmission between layers. It connects all layers directly, with input from previous layers. DenseNet's layers reuse previous layers for efficiency, but repeat unimportant features. The SE block improves DenseNet121 by focusing on channel features with the most information, reducing the impact of unimportant [Albelwi, 2022]

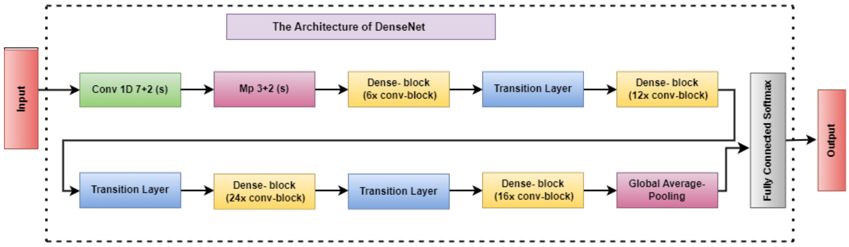

FIG 2.6 – *The Architecture of dense-net*

3.3 Classification

The Softmax classifier is commonly utilized in traditional CNN models to classify input samples into predefined classes, with its input representing the output of the last fully connected layer.[Li et al., 2022].

- The softmax function can be expressed as follows :

$$f_{x_i} = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}}(i = 1, 2, ..., N) \tag{2.1}$$

4 Conclusion

in this chapter, we explored the state-of-the-art techniques in palmprint recognition, focusing on both machine learning and deep learning approaches. Various convolutional neural network (CNN) architectures were examined, including : VGG16,MobilNet,DensNet. In the following chapter, we will present our diverse findings, engaging in both quantitative and qualitative discussions to emphasize the strengths and weaknesses inherent in our system.

Chapter THREE

RESULTS AND DISCUSSIONS

CHAPTER 3

RESULTS AND DISCUSSIONS

1 Introduction

We will present the findings in this chapter to validate the accuracy of our system. We begin by describing the dataset being utilized. Next, using this dataset, we conduct several tests to evaluate our system's impact. Subsequently, we showcase the results obtained from two experiments : transfer learning and fine-tuning. These experiments are conducted on three datasets : right, left, and right+left, using several models including VGG16, DenseNet121, and MobileNetV2. With these results, we can validate the performance of our system.

2 Dataset Description

The PolyU palmprint database is a widely recognized resource in palmprint research. In July 2006, the Biometrics Research Laboratory at IIT Delhi initiated the collection of a touchless palmprint image database to facilitate user-friendly palmprint identification. This database comprises hand images gathered from students and staff at IIT Delhi, New Delhi, India, over a period from July 2006 to June 2007. The database features images from 230 individuals, ranging in age from 12 to 57 years. These images are stored in bitmap format and capture various hand pose variations. For each participant, the images are sequentially numbered using an integer identification system. To enhance usability, the database provides not only the original bitmap images but also automatically cropped and normalized versions. This comprehensive approach makes the PolyU palmprint database a valuable tool for researchers in the field of biometric identification.

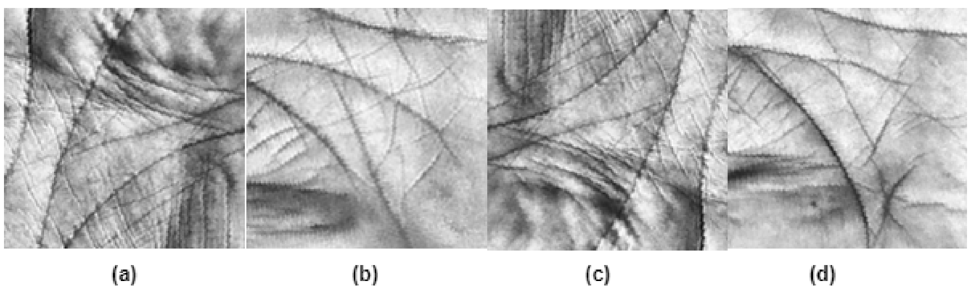

FIG 3.1 – *Some samples from the PolyU palmprint dataset.*

3 Evaluation Metrics

We evaluate the proposed system using several performance metrics : precision, recall, accuracy, and F1-score, as defined in Equations 1, 2, 3, and 4 respectively. These metrics are calculated using the following parameters :

TN : Number of true negative predictions TP : Number of true positive predictions FN : Number of false negative predictions FP : Number of false positive predictions

In addition to these metrics, we plot Cumulative Match Characteristics (CMC) curves to provide a comprehensive overview of each model's performance across different ranks. The Rank-one recognition rate, a key indicator derived from these curves, represents the proportion of probe palmprint images for which the correct identification is returned as the best match from the given set.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}} \tag{3.2}$$

$$\text{Recall} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}} \tag{3.3}$$

$$\text{F1score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.4}$$

4 Results

To achieve good results, we tested our model using various methods. In the first experiment, we employed transfer learning with three models : VGG16, MobileNetV2, and DenseNet121. We fixed the batch size at 32 and the number of epochs at 50. We varied the learning rate across values 0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001, 0.0000001 and used different optimizers (Adam and RMSprop). Once we identified the best learning rate and optimizer, we fixed these parameters and varied the number of epochs 10, 25, 50, 100, 200, 300 and batch size 16, 32, 64, 128. In the second experiment, we used fine-tuning with the same parameters as the first test. The tables below provide examples of these techniques.

4.1 Transfer Learning

Transfer learning is the process of improving learning in a new task by utilizing knowledge previously gained from a related task. While most machine learning algorithms are designed to handle singular tasks, the machine learning community maintains a constant interest in developing algorithms that enable transfer learning [Torrey and Shavlik, 2010].

4.1.1 VGG16 results

- Influence of optimizer and learning rate on the VGG16 performance using transfer learning

In Table 3.1, we set the number of epochs and batch size in advance and start changing the learning rate and the optimizer. For the 'left' data, we obtained the minimum accuracy value of 61.50% at a learning rate of 0.1 and the RMSprop optimizer, a good accuracy value of 77.50%

Datasets	Optimizer	Learning rate							
		0.1	0.01	0.001	0.002	0.004	0.006	0.0001	0.0006
Left	Adam	77.50	78.50	86.00	86.50	86.00	82.50	83.50	88.50
	RMSprop	61.50	81.00	86.50	85.50	83.00	84.5	85.00	89.5
Right	Adam	71.50	76.00	82.50	82.00	83.00	74.00	75.50	79.50
	RMProp	68.50	78.00	81.00	81.00	76.00	75.00	76.00	80.50
Left+Right	Adam	74.25	77.75	85.25	77.00	76.00	77.25	76.50	83.25
	RMSprop	76.25	75.75	85.25	83.25	79.75	77.75	76.50	84.50

TABLE 3.1 – *Results of Optimizers and Learning Rates for Different Datasets using VGG16*

at a learning rate of 0.1 and the Adam optimizer, and the best estimated accuracy value of 89.5% at a learning rate of 0.0006 and the RMSprop optimizer. For the 'right' data, we obtained the worst accuracy value of 68.50% at a learning rate of 0.1 and the RMSprop optimizer, a good accuracy value of 75.50% at a learning rate of 0.001 and the Adam optimizer, and the best estimated accuracy value of 83% at a learning rate of 0.004 and the Adam optimizer. For the 'right+left' multi-instance data, we obtained the worst accuracy value of 74.25% at a learning rate of 0.1 and the Adam optimizer, a good accuracy value of 79.75% at a learning rate of 0.004 and the RMSprop optimizer, and the best estimated accuracy value of 85.25% at a learning rate of 0.001 and the Adam optimizer.

- Influence of epoch number and batch size on the VGG16 performance using transfer learning

Datasets	Epochs number	Batch size			
		16	32	64	128
Left	10	85.50	83.00	81.50	76.50
	20	87.00	85.00	84.50	82.00
	50	87.50	87.50	87.50	83.00
	90	87.00	87.50	89.50	84.50
	100	88.00	89.00	87.50	86.50
	130	90.50	88.50	88.50	87.00
Right	10	74.50	77.00	75.50	76.50
	20	75.50	76.50	76.00	76.00
	50	75.50	80.50	78.50	79.00
	90	76.50	74.50	84.50	80.00
	100	80.00	80.50	81.50	79.50
	130	79.00	78.50	80.50	82.00
Left+Right	10	78.75	78.75	79.00	73.00
	20	81.25	83.25	83.00	77.25
	50	83.50	84.25	84.50	83.25
	90	85.50	85.75	84.75	84.00
	100	83.50	85.75	85.25	84.25
	130	84.25	83.75	86.00	85.50

TABLE 3.2 – *Results of Epoch Numbers and Batch Sizes for Different Datasets using Vgg16*

In Table 3.2, we set the learning rate and the optimizer from the best values in Table 3.1 and start changing the number of epochs and batch size. For the 'left' data, we obtain the worst accuracy value of 76.50% at 10 epochs and a batch size of 128, a good accuracy value of 83% at 10 epochs and a batch size of 32, and the best estimated accuracy value of 90.50% at 130 epochs and a batch size of 16. For the 'right' data, we obtain the worst accuracy value of 74.50% at 10 epochs and a batch size of 16, a good accuracy value of 79.50% at 100 epochs and a batch size of 128, and the best estimated accuracy value of 84.50% at 90 epochs and a batch size of 64. For the 'right+left' multi-instance data, we obtain the worst accuracy value of 73% at 10 epochs and a batch size of 128, a good accuracy value of 79% at 10 epochs and a batch size of 64, and the best estimated accuracy value of 86% at 130 epochs and a batch size of 64.

- Left Dataset
- The values selected are based on the previous tables Model VGG16 Left

Metrics(%)	Learning rate= 0.0006 Optimizer= RMSprop Batch size=16 Number of epochs= 130	Learning rate= 0.01 Optimizer= Adam Batch size= 32 Number of epochs= 50	Learning rate= 0.1 Optimizer= RMSprop Batch size= 32 Number of epochs= 50
R1	89.5	78.5	61.5
R5	96	93	85
Precission	90	79	58
Recall	90	79	61
F1score	88	78	57

TABLE 3.3 – *Values selected VGG16 Left*

In Table 3.3, we utilize four important parameters for evaluating our model : number of epochs, batch size, learning rate, and the optimizer. To begin, we select the best parameters from the previous table. The metrics used for evaluation are F1-score, precision, recall, R1, and R5. Finally, as shown in the table and figures, we record the worst, good, and best results.

- Right Dataset
- The values selected are based on the previous tables Model VGG16 Right

Metrics(%)	Learning rate= 0.004 Optimizer= RMSprop Batch size= 64 Number of epochs= 90	Learning rate= 0.0001 Optimizer= RMSprop Batch size= 32 Number of epochs= 50	Learning rate= 0.001 Optimizer= RMSprop Batch size= 32 Number of epochs= 50
R1	84.50	75.5	68.5
R5	92	89	91
Precission	85	77	74
Recall	84	76	69
F1score	83	74	67

TABLE 3.4 – *Values selected VGG16 Right*

In Table 3.4, we utilize four important parameters for evaluating our model : number of epochs, batch size, learning rate, and the optimizer. To begin, we select the best parameters from the previous tables. The evaluation metrics used are F1-score, precision, recall, R1, and R5. Finally, as shown in the table and figures, we record the worst, good, and best results.

- Left+Right Dataset

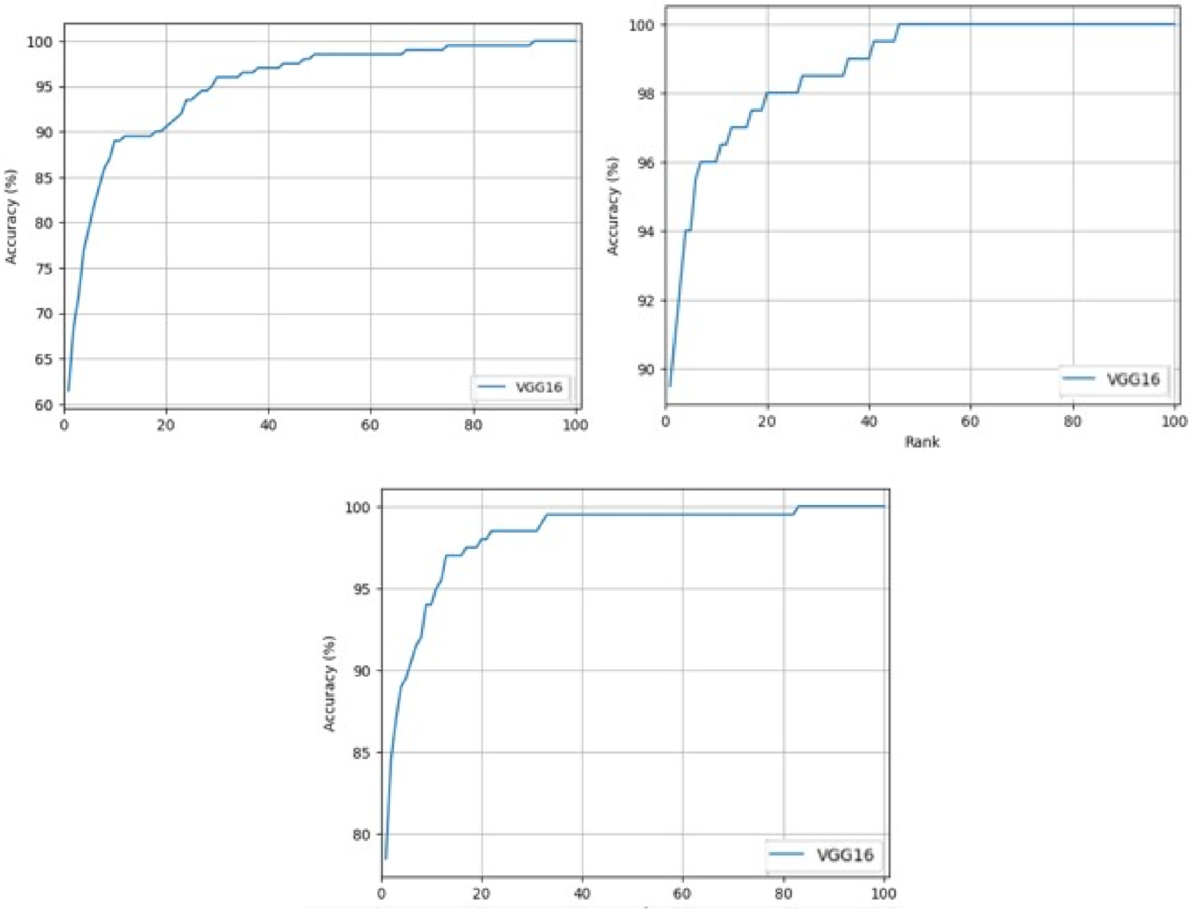

FIG 3.2 – *CMC curve using Vgg16 on left dataset*

Metrics(%)	Learning rate= 0.001 Optimizer= Adam Batch size= 32 Number of epochs=50	Learning rate=0.001 Optimizer= Adam Batch size=64 Number of epochs=10	Learning rate= 0.001 Optimizer= Adam Batch size=128 Number of epochs=10
R1	85.25	79	73
R5	92	94	95
Precission	88	83	82
Recall	85	79	74
F1score	85	78	74

TABLE 3.5 – *Values selected VGG16 Left+Right*

- The values selected are based on the previous tables Model VGG16 Left+Right

In Table 3.5, we utilize four important parameters for evaluating our model : number of epochs, batch size, learning rate, and the optimizer. To begin, we select the best parameters from the previous tables. The evaluation metrics used are F1-score, precision, recall, R1, and R5. Finally, as shown in the table and figures, we record the worst, good, and best results.

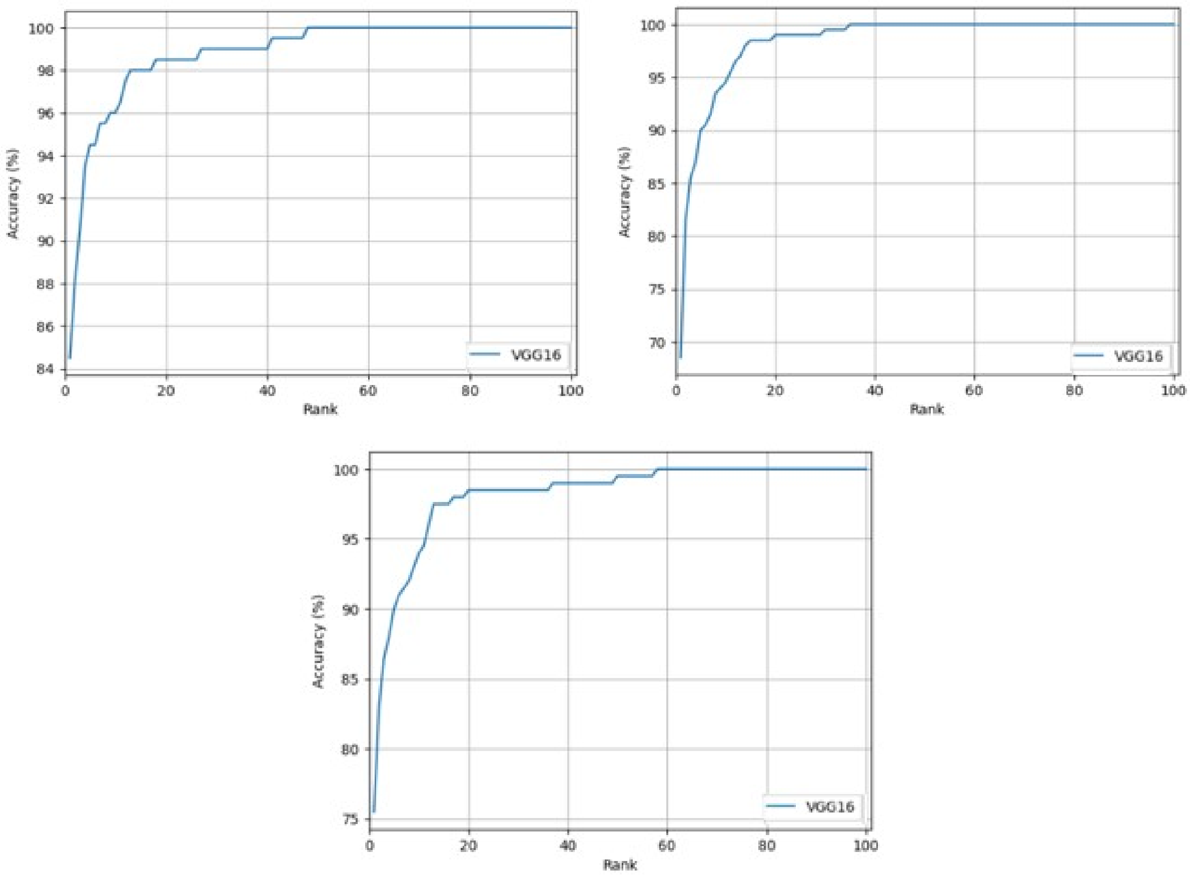

FIG 3.3 – *CMC curve using Vgg16 on Right dataset*

4.1.2 DenseNet-121 results

- Influence of optimizer and learning rate on the DenseNet-121 performance transfer learning

Datasets	Optimizer	Learning rate							
		0.1	0.01	0.001	0.002	0.004	0.006	0.0001	0.0006
Left	Adam	67.50	88.50	87.00	88.00	89.50	90.50	78.00	88.50
	RMSprop	77.00	86.00	90.00	87.50	49.50	88.50	81.00	87.00
Right	Adam	75.00	85.00	87.00	88.50	87.50	88.00	77.50	87.50
	RMSprop	66.50	81.00	89.00	88.50	86.50	86.50	77.50	87.00
Left+Right	Adam	50.5	88.00	86.75	87.00	86.00	87.50	78.25	85.50
	RMSprop	69.25	76.00	86.75	86.25	87.25	83.00	78.25	86.50

TABLE 3.6 – *Results of Optimizers and Learning Rates for Different Datasets using DenseNet-121*

In Table 3.6, we set the number of epochs and batch size in advance and start changing the learning rate and the optimizer. For the 'left' data, we obtain the worst accuracy value of 67.50% at a learning rate of 0.1 and the Adam optimizer, a good accuracy value of 78% at a learning rate of 0.0001 and the Adam optimizer, and the best estimated accuracy value of 90.50% at a learning rate of 0.006 and the Adam optimizer. For the 'right' data, we obtain the worst accuracy value of 66.50% at a learning rate of 0.1 and the RMSprop optimizer, a good accuracy value of

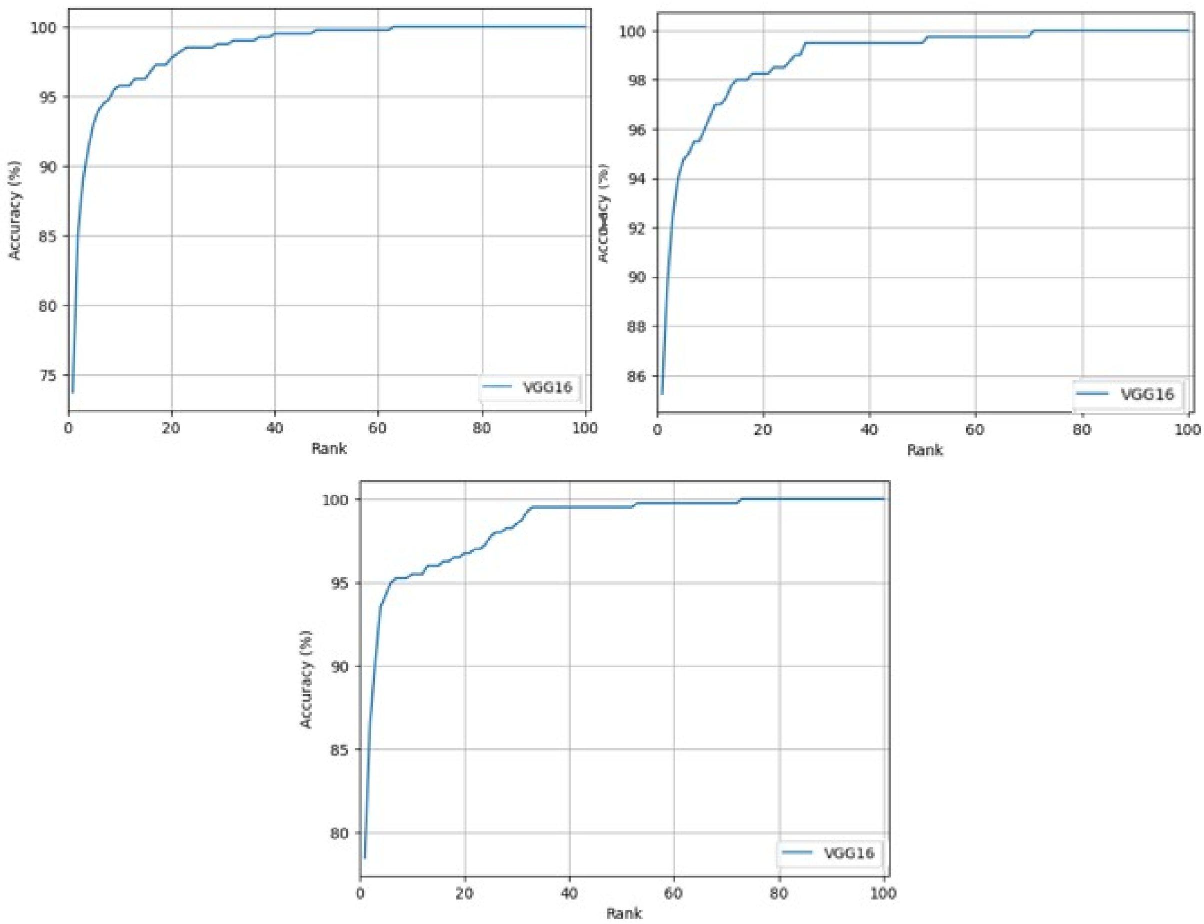

FIG 3.4 – *CMC curve using Vgg16 on Right+Left dataset*

77.50% at a learning rate of 0.0001 and the Adam optimizer, and the best estimated accuracy value of 89% at a learning rate of 0.001 and the RMSprop optimizer. For the 'right+left' multi-instance data, we obtain the worst accuracy value of 50.50% at a learning rate of 0.1 and the Adam optimizer, a good accuracy value of 69.25% at a learning rate of 0.1 and the RMSprop optimizer, and the best estimated accuracy value of 88% at a learning rate of 0.01 and the Adam optimizer.

- Influence of epoch number and batch size on the DenseNet-121 performance using transfer learning

In Table 3.7, we set the learning rate and the optimizer from the best values in Table 3.6 and start changing the number of epochs and batch size. For the 'left' data, we obtain the worst accuracy value of 64.50% at 10 epochs and a batch size of 128, a good accuracy value of 80% at 20 epochs and a batch size of 128, and the best estimated accuracy value of 91.50% at 90 epochs and a batch size of 32. For the 'right' data, we obtain the worst accuracy value of 64.50% at 20 epochs and a batch size of 128, a good accuracy value of 75% at 10 epochs and a batch size of 32, and the best estimated accuracy value of 90.50% at 130 epochs and a batch size of 16. For the 'right+left' multi-instance data, we obtain the worst accuracy value of 69.50% at 20 epochs and a batch size of 16, a good accuracy value of 77% at 90 epochs and a batch size of 16, and the best estimated accuracy value of 87% at 90 epochs and a batch size of 32.

- Left Dataset

Datasets	Epochs number	Batch size			
		16	32	64	128
Left	10	87.00	88.00	82.50	64.50
	20	88.50	89.50	87.50	80.00
	50	90.00	88.50	86.50	87.00
	90	90.50	91.50	88.50	85.50
	100	89.50	90.50	88.00	87.00
	130	89.50	88.50	87.00	90.00
Right	10	84.00	75.00	65	50
	20	87.50	87.00	83.50	64.50
	50	86.50	86.50	87.50	81.50
	90	89.50	87.00	85.50	84.50
	100	90.00	88.00	85.50	87.00
	130	90.50	88.00	88.50	86.00
Left+Right	10	85.50	86.75	85.25	80.75
	20	69.50	86.50	86.00	85.75
	50	81.75	86.00	86.75	84.75
	90	77.00	87.00	86.50	86.25
	100	83.25	89.00	87.50	84.75
	130	82.75	89.00	87.25	86.75

TABLE 3.7 – *Results of Epoch Numbers and Batch Sizes for Different Datasets using DenseNet-121*

- The values selected are based on the previous tables Model DenseNet-121 Left

Metrics(%)	Learning rate= 0.006 ptimizer= Adam Batch size=32 Number of epochs= 90	Learning rate= 0.1 Optimizer= RMSprop Batch size= 32 Number of epochs= 50	Learning rate= 0.005 Optimizer= Adam Batch size= 128 Number of epochs= 10
R1	91.5	77	64.5
R5	97	90.3	95
Precission	92	76	76
Recall	92	74	78
F1score	91	72	74

TABLE 3.8 – *Values selected DenseNet-121 on Left*

In Table 3.8,we use the four parameters that are important for evaluating our model :number of epochs , batch size, learning rate and the optimizer. To begin, we select the best parameters from the previous table. The F1-score, precision, recall,R1 and R5. Finally, as shown in the table and figures, we save the worst , good and best results.

- Right Dataset
- The values selected are based on the previous tables Model denseNet-121 Right

In Table 3.9,we use the four parameters that are important for evaluating our model : number of epochs , batch size, learning rate and the optimizer. To begin, we select the best parameters

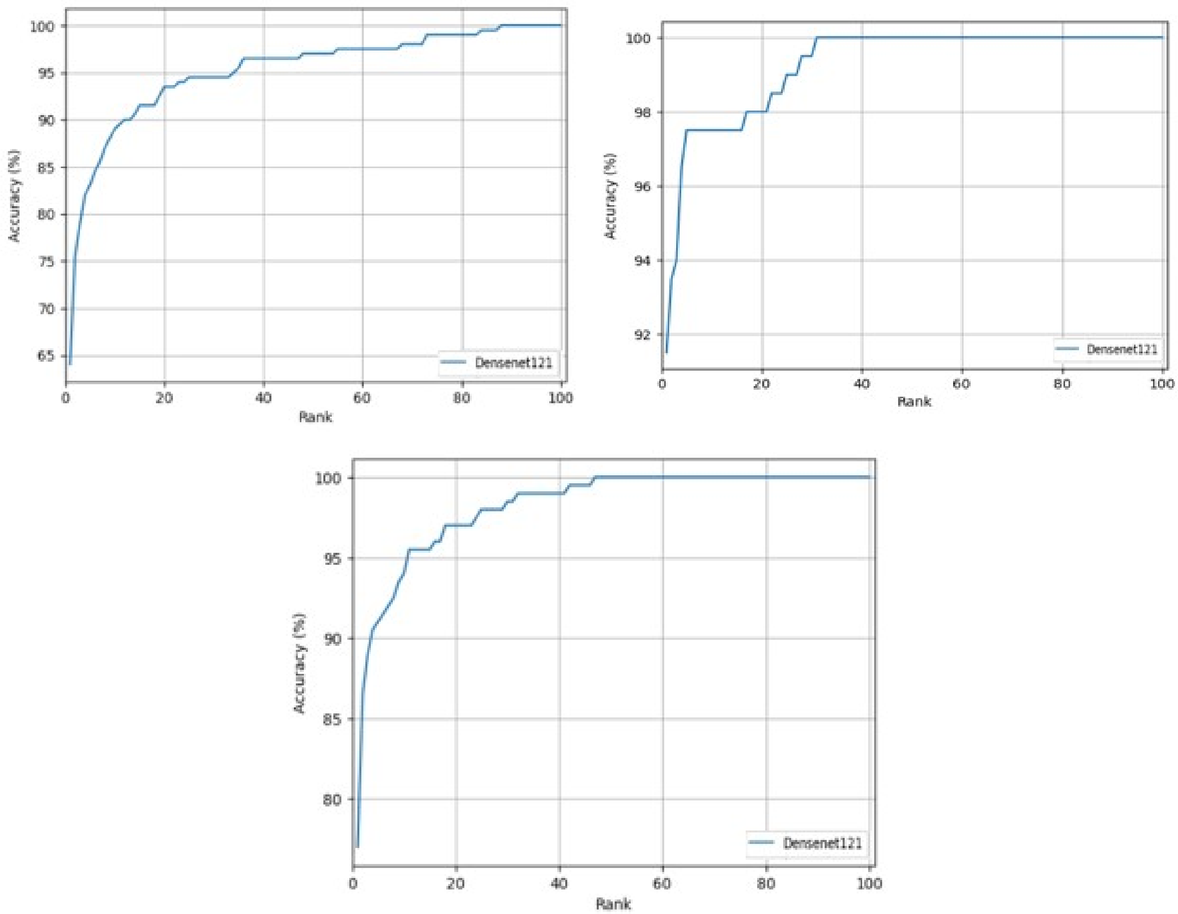

FIG 3.5 – *CMC curve using denseNet-121 on left dataset*

Metrics(%)	Learning rate= 0.001 Optimizer= RMSprop Batch size= 16 Number of epochs= 130	Learning rate= 0.0001 Optimizer= RMSprop Batch size= 32 Number of epochs= 50	Learning rate= 0.001 Optimizer= RMSprop Batch size= 128 Number of epochs= 20
R1	90.50	77.50	64.50
R5	95	92	93
Precission	95	77	68
Recall	91	77	65
F1score	91	74	62

TABLE 3.9 – *Values selected DenseNet-121 Right*

from the previous table. The F1-score, precision, recall,R1 and R5. Finally, as shown in the table and figures, we save the worst , good and best results.

- Left+Right Dataset
- The values selected are based on the previous tables Model DenseNet-121 Left+Right

In Table 3.10,we use the four parameters that are important for evaluating our model : number of epochs , batch size, learning rate and the optimizer. To begin, we select the best parameters from the previous table. The F1-score, precision, recall,R1 and R5. Finally, as shown in the table and figures, we save the worst , good and best results.

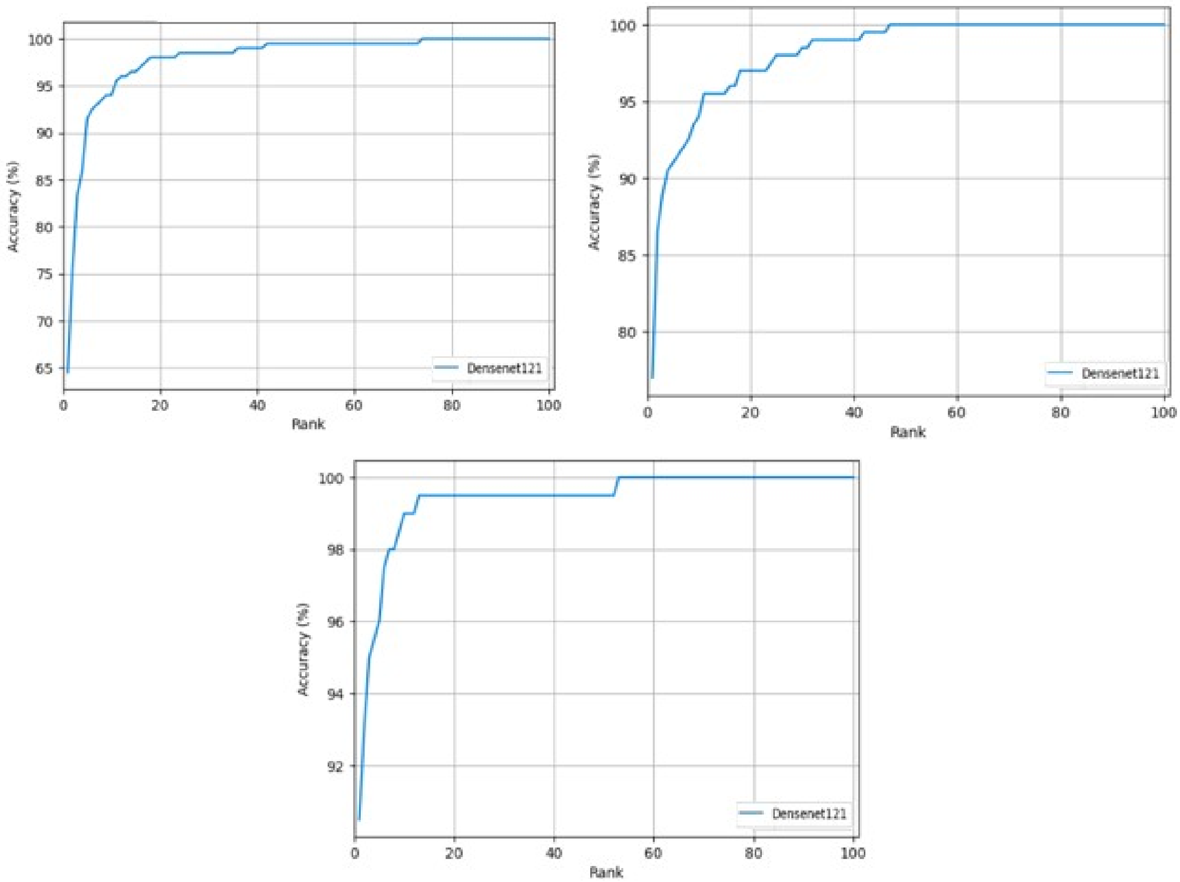

FIG 3.6 – *CMC curve using DenseNet-121 on Right dataset*

Metrics(%)	Learning rate= 0.01 Optimizer= Adam Batch size= 32 Number of epochs=90	Learning rate= 0.01 Optimizer= Adam Batch size= 16 Number of epochs=20	Learning rate= 0.1 Optimizer= Adam Batch size= 32 Number of epochs=50
R1	87.00	69.50	50.50
R5	94	95	96
Precission	89	88	89
Recall	87	85	88
F1score	87	85	87

TABLE 3.10 – *Values selected DenseNet-121 Left+Right*

4.1.3 MobileNet-V2 results

- Influence of optimizer and learning rate on the MobileNet-V2 performance using transfer learning

In Table 3.11, we set the number of epochs and batch size in advance and start changing learning rate and the optimizer,in data left until we get the worst value accuracy of 68.50% at learning rate 0.1 and optimizer Adam ,the good value accuracy 78.50% at learning rate 0.01 and optimizer RMSprop and best estimated value accuracy of 90% at learning rate 0.004 and optimizer Adam , in data right until we get the worst value accuracy of 64% at learning rate 0.1 and optimizer Adam, the good value accuracy of 76% at learning rate 0.001 and optimizer

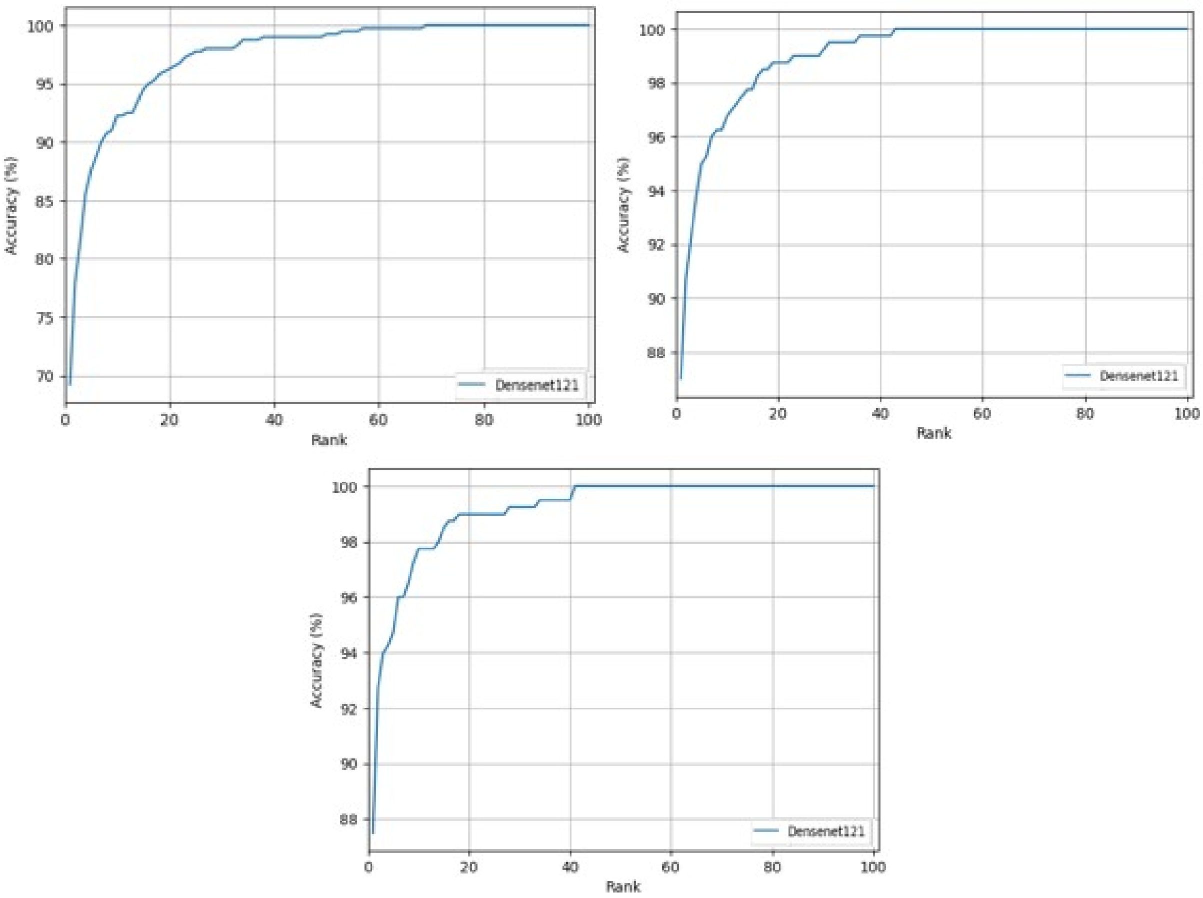

FIG 3.7 – *CMC curve using DenseNet-121 on Left+Right dataset*

Datasets	Optimizer	Learning rate							
		0.1	0.01	0.001	0.002	0.004	0.006	0.0001	0.0006
Left	Adam	68.50	88.00	88.50	89.50	90.00	88.00	84.50	88.50
	RMSprop	68.50	78.50	89.00	89.00	79.00	84.50	86.00	86.50
Right	Adam	64.00	82.00	82.00	84.50	83.50	84.00	81.00	83.00
	RMSprop	72.00	76.00	82.00	69.00	81.00	78.00	82.00	82.00
Left+Right	Adam	50.50	84.50	82.50	82.00	84.50	83.50	80.00	83.00
	RMSprop	60.00	69.50	82.50	81.00	81.50	80.50	79.00	82.50

TABLE 3.11 – *Results of Optimizers and Learning Rates for Different Datasets using MobileNet-V2*

RMSprop and best estimated value accuracy of 84.50% at learning rate 0.002and optimizer Adam. In multi-instance until we get the worst value accuracy of 50.50% at learning rate 0.1 and optimizer Adam ,the good value accuracy of 69.50% at learning rate 0.01 and optimizer RMSprop and best estimated value accuracy of 84.50% at learning rate 0.004 and optimizer Adam.

- Influence of epoch number and batch size on the MobileNet-V2 performance using transfer learning

In Table 3.12, we set learning rate and the optimizer from the best value in Table 3.11 and start changing the number of epochs and batch size , in data left until we get the worst value

Datasets	Epochs number	Batch size			
		16	32	64	128
Left	10	85.50	86.50	82.00	65.00
	20	90.00	87.50	88.50	85.00
	50	88.00	89.00	89.00	79.50
	90	90.00	88.00	88.50	86.00
	100	89.50	90.50	87.50	86.00
	130	82.00	88.50	88.00	87.00
Right	10	83.00	83.50	74.50	70
	20	82.0	82.50	83.00	78.50
	50	85.00	83.00	81.50	82.50
	90	83.50	83.50	82.50	83.00
	100	83.50	84.50	83.00	81.00
	130	86.00	84.00	83.00	81.50
Left+Right	10	80.50	78.75	78.00	70.75
	20	86.25	83.25	83.00	77.25
	50	82.50	84.25	84.50	83.25
	90	85.50	85.75	84.75	84.00
	100	80.50	85.75	85.25	84.25
	130	84.25	83.75	86.00	85.50

TABLE 3.12 – *Results of Epoch Numbers and Batch Sizes for Different Datasets using MobileNet-V2*

accuracy of 65% at numbres epcohs 10 and batch size 128, ,the good value accuracy 79.50%at numbres epcohs 50 and batch size 128 and best estimated value accuracy of 90.50% at numbres epcohs 100 and batch size 32. in data right until we get the worst value accuracy of 74.50% at numbres epcohs 10 and batch size 64, ,the good value accuracy of 81% at numbres epcohs 100 and batch size 128 and best estimated value accuracy of 86% at numbres epcohs 130 and batch size 16.In multi-instance until we get the worst value accuracy of 70.75% at numbres epcohs 10 and batch size 128, ,the good value accuracy of 78.50% at numbres epcohs 10 and batch size 32 and best estimated value accuracy of 86.25% at numbres epcohs 10 and batch size 16.

- Left Dataset
- The values selected are based on the previous tables Model MobileNet-V2 Left

Metrics(%)	Learning rate= 0.005 Optimizer= Adam Batch size=32 Number of epochs= 100	Learning rate= 0.001 Optimizer= RMSprop Batch size= 32 Number of epochs= 50	Learning rate= 0.005 Optimizer= Adam Batch size= 128 Number of epochs= 10
R1	90.5	78.5	65
R5	95	97	85
Precission	92	89	71
Recall	91	86	65
F1score	90	85	65

TABLE 3.13 – *Values selected MobileNet-V2 on Left*

In Table 3.13,we use the four parameters that are important for evaluating our model :number of epochs , batch size, learning rate and the optimizer. To begin, we select the best parameters from the previous table. The F1-score, precision, recall,R1 and R5. Finally, as shown in the table and figures, we save the worst , good and best results.

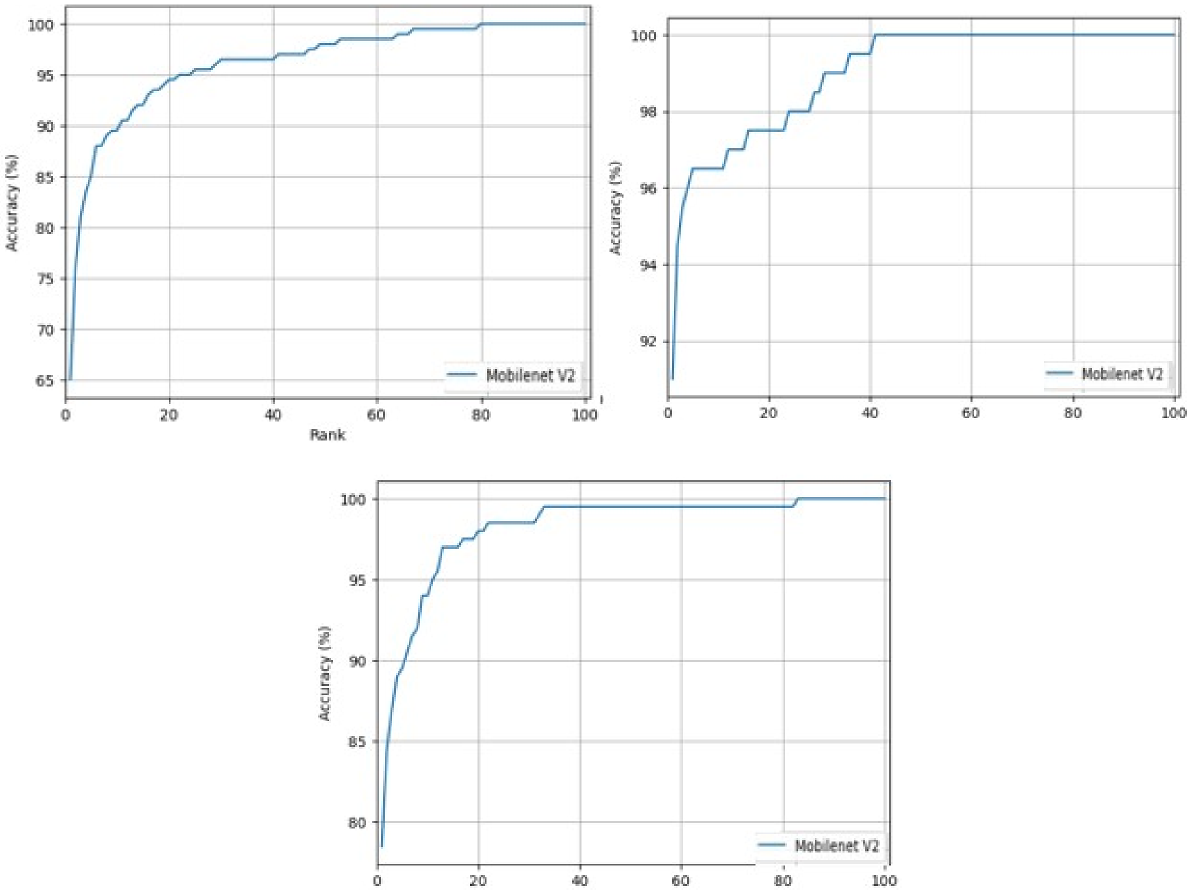

FIG 3.8 – *CMC curve using MobileNet-V2 on left dataset*

- Right Dataset
- The values selected are based on the previous tables Model MobileNet-V2 Right

Metrics(%)	Learning rate= 0.002 Optimizer= Adam Batch size= 16 Number of epochs= 130	Learning rate= 0.002 Optimizer= Adam Batch size= 64 Number of epochs= 10	Learning rate= 0.1 Optimizer= Adam Batch size= 32 Number of epochs= 50
R1	86	74.5	64
R5	93	87	82
Precission	90	79	70
Recall	86	74	64
F1score	85	73	63

TABLE 3.14 – *Values selected MobileNet-V2 Right*

In Table 3.14,we use the four parameters that are important for evaluating our model :number of epochs , batch size, learning rate and the optimizer. To begin, we select the best parameters

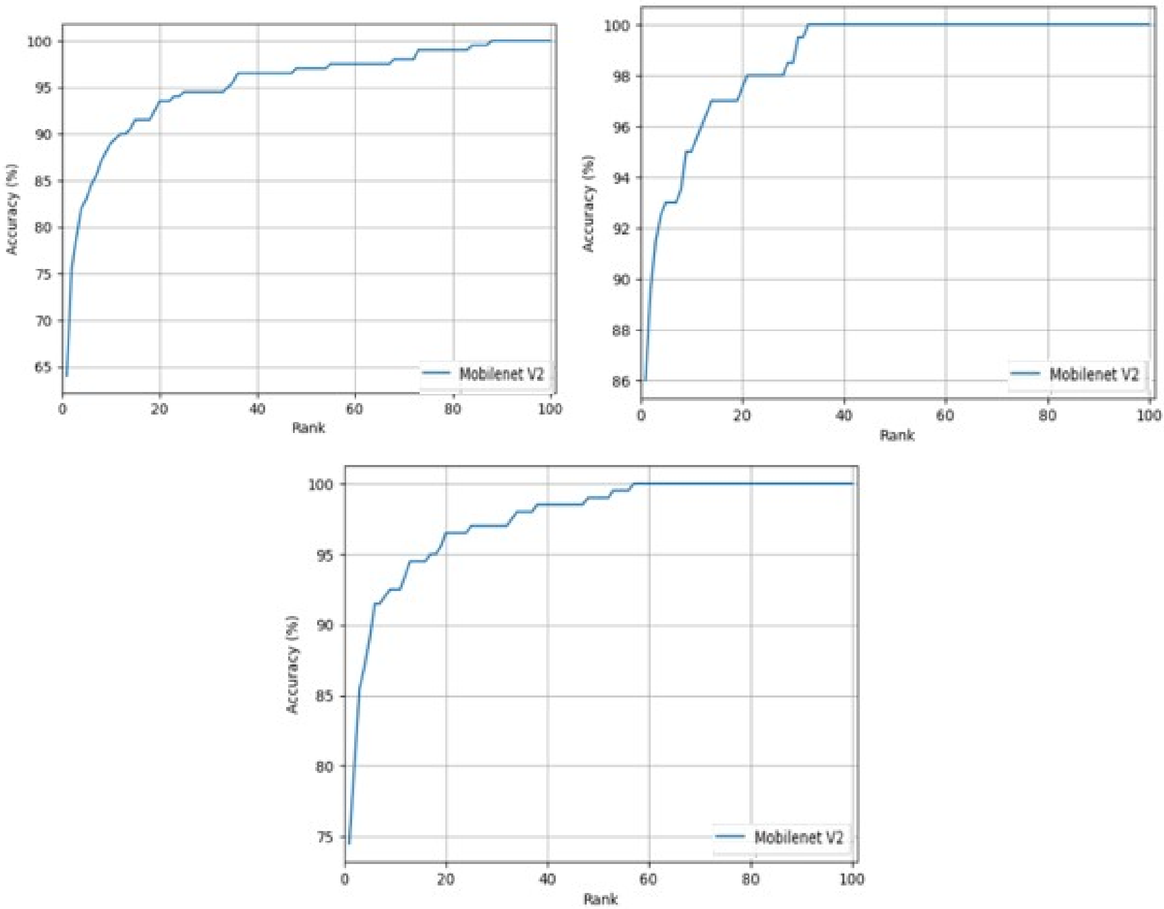

FIG 3.9 – *CMC curve using MobileNet-V2 on Right dataset*

from the previous table. The F1-score, precision, recall,R1 and R5. Finally, as shown in the table and figures, we save the worst , good and best results.

- Left+Right Dataset
- The values selected are based on the previous tables Model MobileNet-V2 Left+Right

Metrics(%)	Learning rate= 0.004 Optimizer= Adam Batch size=16 Number of epochs=20	Learning rate= 0.01 Optimizer= RMSprop Batch size=16 Number of epochs=50	Learning rate=0.1 Optimizer= Adam Batch size=32 Number of epochs=50
R1	86.25	69.50	50.5
R5	96.4	94	91
Precission	89	86	79
Recall	86	82	76
F1score	86	81	74

TABLE 3.15 – *Values selected MobileNet-V2 Left+Right*

In Table 3.15,we use the four parameters that are important for evaluating our model : number of epochs , batch size, learning rate and the optimizer. To begin, we select the best parameters from the previous table. The F1-score, precision, recall,R1 and R5. Finally, as shown in the table and figures, we save the worst , good and best results.

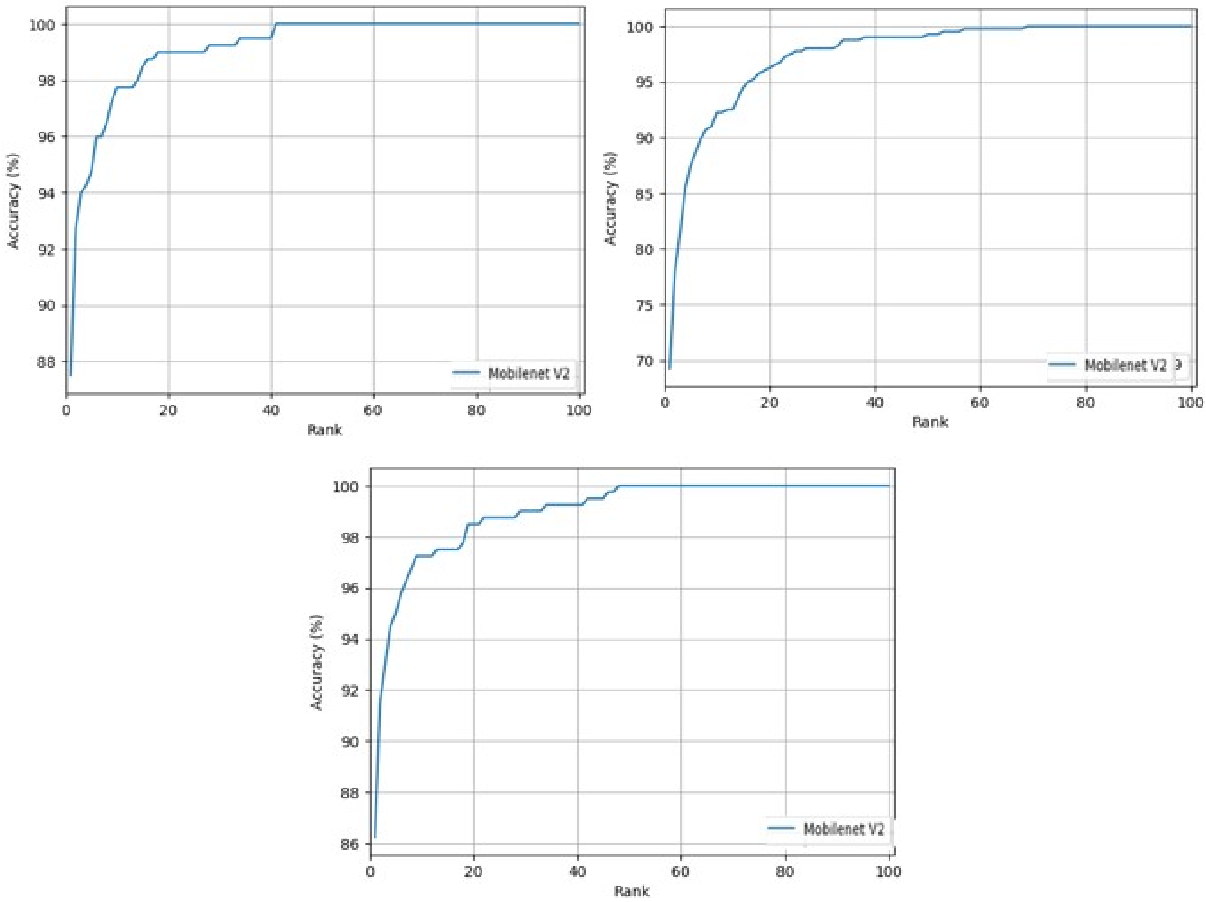

FIG 3.10 – *CMC curve using MobileNet-V2 on Left+Right dataset*

4.2 Fine-tuning

The fine-tuning approach is one of the most popular transfer learning strategies used in neural network applications. This method, which involves transferring knowledge from a generative model to a discriminative model, was pioneered to achieve high generalization[Vrbančič and Podgorelec, 2020]

4.2.1 VGG16 results

- Influence of optimizer and learning rate on the VGG16 performance using fine tuning technique

Datasets	Optimizer	Learning rate							
		0.01	0.001	0.002	0.0004	0.005	0.0005	0.006	0.0006
Left	Adam	65	82	80	89	88	91	91	89
	RMSprop	70	85	88	87	90	88	82	81
Right	Adam	63	81	83	82	89	87	82	79
	RMSprop	89	91	90	89	89	88	92	90
Left+Right	Adam	84	86	89	90	88	90	88	88
	RMSprop	85	89	84	91	90	88	89	87

TABLE 3.16 – *Results of Optimizers and Learning Rates for Different Datasets using VGG16*

In Table 3.16, we set the number of epochs and batch size in advance and start changing learning rate and the optimizer, in data left until we get the worst value accuracy of 65% at learning rate 0.01and optimizer Adam,the good value accuracy 80% at learning rate 0.002 and optimizer Adam and best estimated value accuracy of 91% at learning rate 0.0005 and optimizer Adam , in data right until we get the worst value accuracy of 63% at learning rate 0.01 and optimizer Adam, the good value accuracy of 81% at learning rate 0.001 and optimizer Adam and best estimated value accuracy of 92% at learning rate 0.006% and optimizer RMSprop.In multi-instance until we get the worst value accuracy of 84% at learning rate 0.01 and optimizer Adam,the good value accuracy of 87% at learning rate 0.0006 and optimizer RMSprop and best estimated value accuracy of 91% at learning rate 0.004 and optimizer RMSprop.

- Influence of epoch number and batch size on the VGG16 performance using fine tuning technique

Datasets	Epochs number	Batch size			
		16	32	64	128
Left	10	89	87	80	88
	25	88	89	83	87
	35	90	91	87	81
	40	92.5	92	90	82
	50	91	92	89	83
Right	10	92	89	92	88
	25	92	90	90	87
	35	94	90	89	89
	40	90	91	90	92
	50	93	93	91	91
Left+Right	10	92	89	94	90
	25	90	92	94	92
	35	88	93	96.50	91
	40	87	95	95	94
	50	92	95	96	94

TABLE 3.17 – *Results of Epoch Numbers and Batch Sizes for Different Datasets using Vgg16*

In Table 3.17, we set learning rate and the optimizer from the best value in Table 3.16 and start changing the number of epochs and batch size , in data left until we get the worst value accuracy of 81% at numbres epcohs 35 and batch size 128, ,the good value accuracy 87% at numbres epcohs 10 and batch size 32 and best estimated value accuracy of 92.5% at numbres epcohs 40 and batch size 16,in data right until we get the worst value accuracy of 87% at numbres epcohs 25 and batch size 128, ,the good value accuracy of 90% at numbres epcohs 40 and batch size 16 and best estimated value accuracy of 94% at numbres epcohs 35and batch size 16. In multi-instance until we get the worst value accuracy of 87% at numbres epcohs 40 and batch size 16, ,the good value accuracy of 92% at numbres epcohs 10 and batch size 16 and best estimated value accuracy of 96.5% at numbres epcohs 35and batch size 64.

- Left Dataset
- The values selected are based on the previous tables Model VGG16 Left

In Table 3.18,,we use the four parameters that are important for evaluating our model : number of epochs , batch size, learning rate and the optimizer. To begin, we select the best parameters from the previous table. The F1-score, precision, recall,R1 and R5. Finally, as shown in the table and figures, we save the worst , good and best results.

Metrics(%)	Learning rate= 0.01 Optimizer= Adam Batch size=32 Number of epochs= 50	Learning rate= 0.01 Optimizer= RMSprop Batch size=32 Number of epochs= 50	Learning rate= 0.006 Optimizer= RMSprop Batch size= 16 Number of epochs= 40
R1	65	89	92.50
R5	83	93	97.00
Precission	66	91	95
Recall	65	89	94
F1score	61	88	93

TABLE 3.18 – *Values selected VGG16 Left*

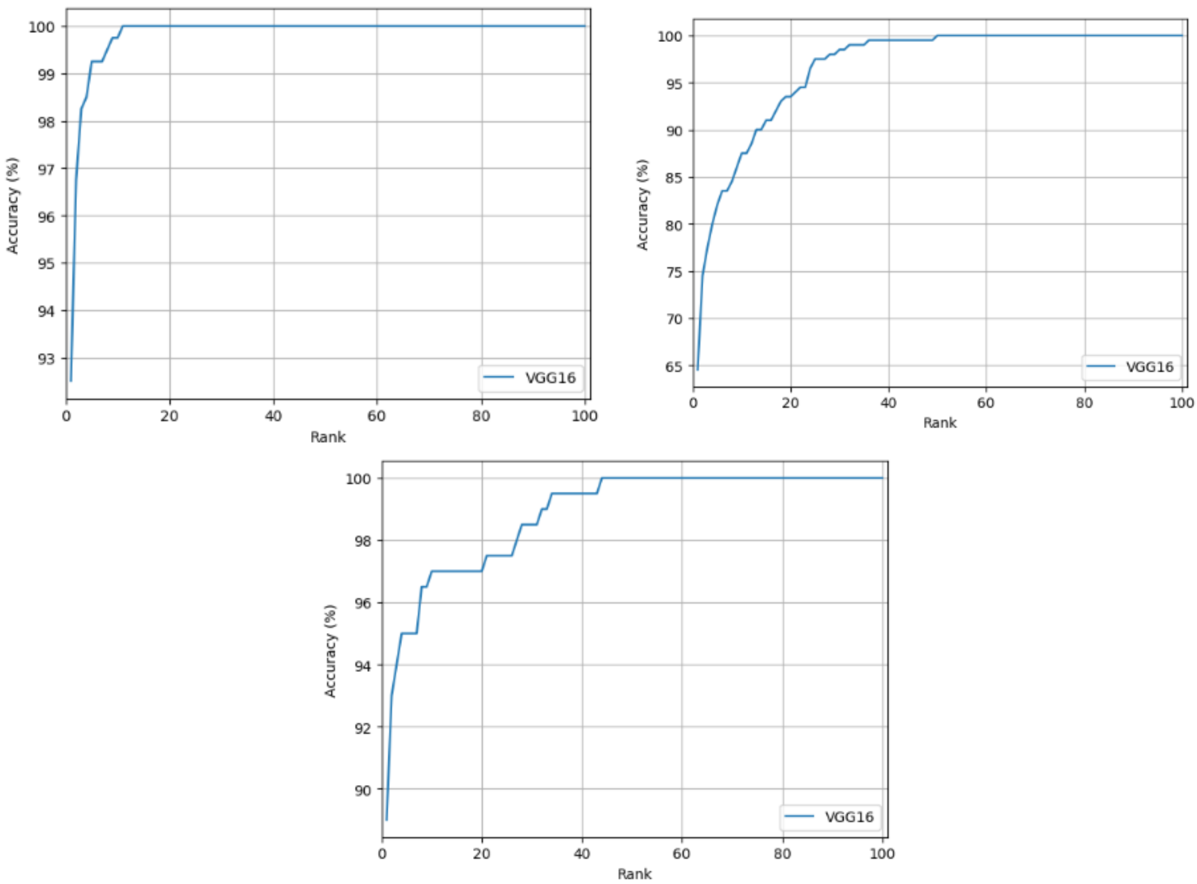

FIG 3.11 – *CMC curve using Vgg16 on left dataset*

- Right Dataset
- The values selected are based on the previous tables Model VGG16 Right

In Table 3.19,we use the four parameters that are important for evaluating our model : number of epochs , batch size, learning rate and the optimizer. To begin, we select the best parameters from the previous table. The F1-score, precision, recall,R1 and R5. Finally, as shown in the table and figures, we save the worst , good and best results.

- Left+Right Dataset
- The values selected are based on the previous tables Model VGG16 Left+Right

Metrics(%)	Learning rate= 0.01 Optimizer= RMSprop Batch size=32 Number of epochs= 50	Learning rate= 0.006 Optimizer= RMSprop Batch size=32 Number of epochs= 40	Learning rate= 0.006 Optimizer= RMSprop Batch size= 16 Number of epochs= 30
R1	63	91	94.00
R5	84	96	98.50
Precission	64	94	95
Recall	63	91	94
F1score	59	91	94

TABLE 3.19 – *Values selected VGG16 Right*

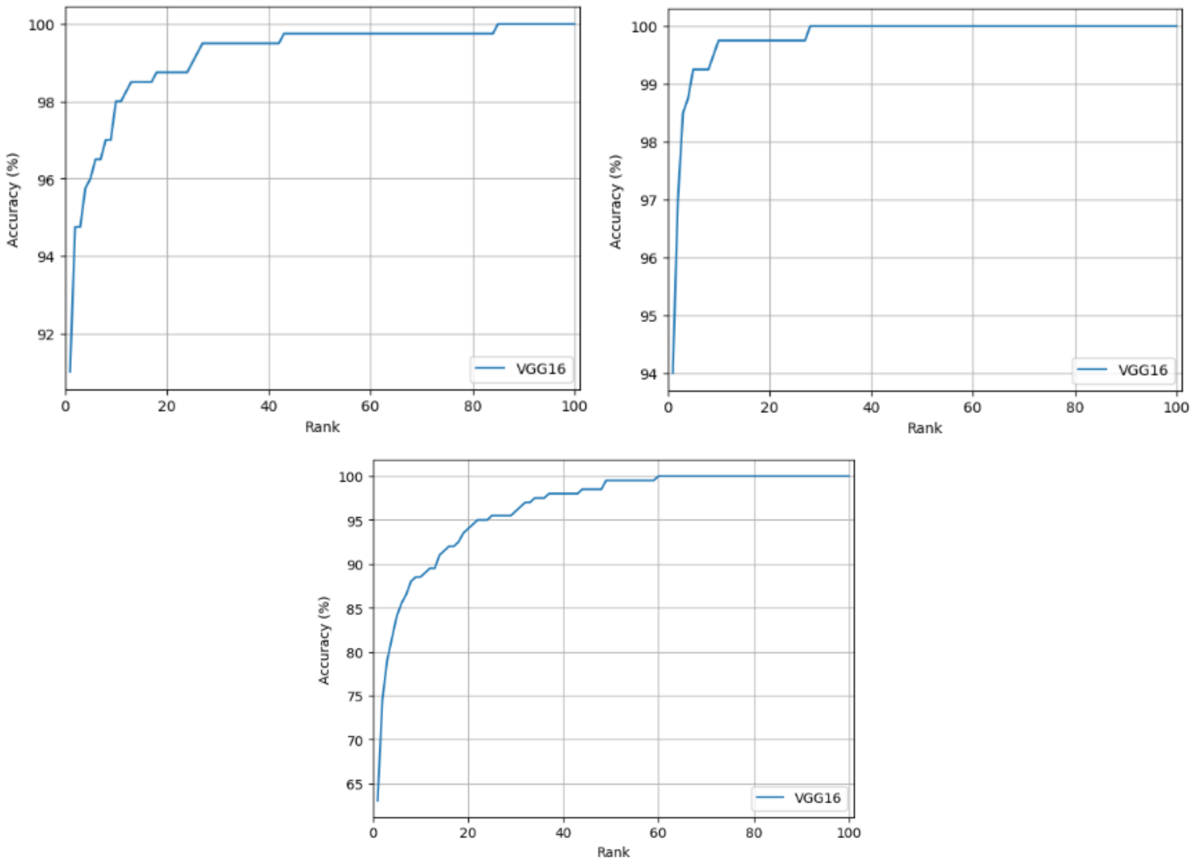

FIG 3.12 – *CMC curve using Vgg16 on Right dataset*

In Table 3.20,we use the four parameters that are important for evaluating our model : number of epochs , batch size, learning rate and the optimizer. To begin, we select the best parameters from the previous table. The F1-score, precision, recall,R1 and R5. Finally, as shown in the table and figures, we save the worst , good and best results.

4.2.2 DenseNet-121 results

- Influence of optimizer and learning rate on the DenseNet-121 performance using fine tuning technique

In Table 3.21, we set the number of epochs and batch size in advance and start changing lear-

Metrics(%)	Learning rate= 0.001 Optimizer= RMSprop Batch size=16 Number of epochs= 35	Learning rate= 0.005 Optimizer= RMSprop Batch size=32 Number of epochs= 50	Learning rate= 0.004 Optimizer= RMSprop Batch size= 64 Number of epochs= 35
R1	84	90	96.50
R5	93	96	99
Precission	91	94	97
Recall	83	90	96
F1score	85	91	96

TABLE 3.20 – *Values selected VGG16 Left+Right*

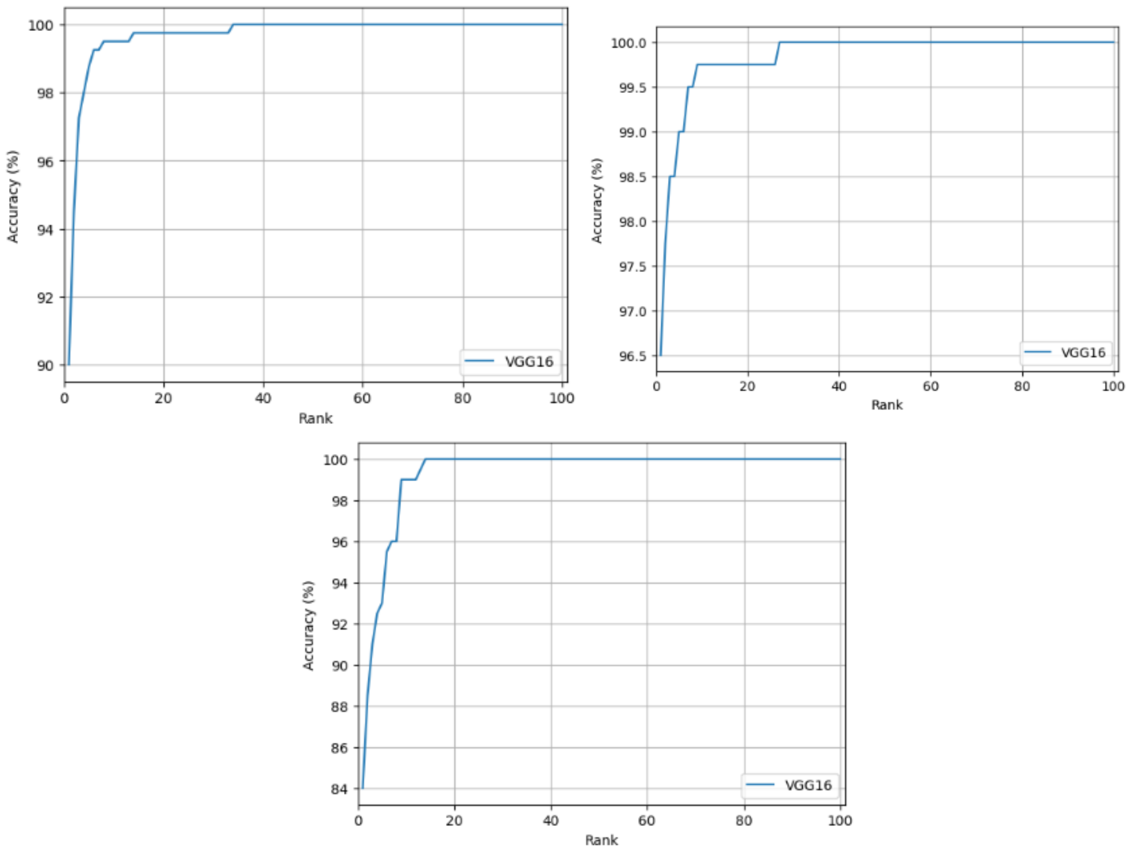

FIG 3.13 – *CMC curve using Vgg16 on Right+Left dataset*

ning rate and the optimizer,in data left until we get the worst value accuracy of 88% at learning rate 0.01and optimizer Adam ,the good value accuracy 90% at learning rate 0.001and optimizer Adam and best estimated value accuracy of 92% at learning rate 0.002and optimizer RMSprop, ,in data right until we get the worst value accuracy of 65% at learning rate 0.01and optimizer Adam, the good value accuracy of 80% at learning rate 0.001and optimizer Adam and best estimated value accuracy of 92% at learning rate 0.0005%and optimizer RMSprop. In multi-instance until we get the worst value accuracy of 83% at learning rate 0.01and optimizer Adam , ,the good value accuracy of 87% at learning rate 0.001and optimizer RMSprop and best estimated value accuracy of 92% at learning rate 0.0006and optimizer RMSprop.

- Influence of epoch number and batch size on the DenseNet-121 performance using fine tuning technique

Datasets	Optimizer	Learning rate							
		0.01	0.001	0.002	0.0004	0.005	0.0005	0.006	0.0006
Left	Adam	88	90	91	90	89	89.5	88	88
	RMSprop	88.5	90	92	92	89	90	91	91
Right	Adam	65	80	85	85	89	90	90	89
	RMSprop	87	85	84	87	90	92	92	91
Left+Right	Adam	83	88	89	88	90	89	91	90
	RMSprop	84	87	90	88	90	87	90	92

TABLE 3.21 – *Results of Optimizers and Learning Rates for Different Datasets using DenseNet-121*

Datasets	Epochs number	Batch size			
		16	32	64	128
Left	10	92	90	93	89
	25	95	89	94.5	89
	35	97.5	91	92	92
	40	95	94	92	91
	50	97	94	92	93
Right	10	90	93	90	91.5
	25	90	92	92	92
	35	91.5	96	95	89
	40	93	96	96.5	90
	50	95	96	96	91
Left+Right	10	89	92	93	90
	25	92	94	95	92
	30	90	92.25	98	92
	40	91	93	96	91
	50	94	98.75	97	93

TABLE 3.22 – *Results of Epoch Numbers and Batch Sizes for Different Datasets using DenseNet-121*

In Table 3.22, we set learning rate and the optimizer from the best value in Table 3.21 and start changing the number of epochs and batch size , in data left until we get the worst value accuracy of 89% at numbres epcohs 10 and batch size 128, ,the good value accuracy 83%at numbres epcohs 50 and batch size 128 and best estimated value accuracy of 97.5% at numbres epcohs 35 and batch size 128,in data right until we get the worst value accuracy of 89% at numbres epcohs 35 and batch size 128, ,the good value accuracy of 93%at numbres epcohs 40 and batch size 16 and best estimated value accuracy of 96.5% at numbres epcohs 40 and batch size 64.In multi-instance until we get the worst value accuracy of 89% at numbres epcohs 10 and batch size 16, ,the good value accuracy of 93%at numbres epcohs 40 and batch size 64 and best estimated value accuracy of 98.75% at numbres epcohs 50 and batch size 32.

- Left Dataset
- The values selected are based on the previous tables Model DenseNet-121 Left

In Table 3.23,we use the four parameters that are important for evaluating our model : number of epochs , batch size, learning rate and the optimizer. To begin, we select the best parameters

Metrics(%)	Learning rate=0.01 Optimizer=RMSprop Batch size=32 Number of epochs=50	Learning rate=0.006 Optimizer=RMSprop Batch size=16 Number of epochs=25	Learning rate=0.006 Optimizer=RMSprop Batch size=16 Number of epochs=35
R1	88.50	94.5	97.50
R5	96	97	99
Precision	90	95	97
Recall	89	94	97
F1 score	88	94	97

TABLE 3.23 – *Values selected DenseNet-121 on Left*

from the previous table. The F1-score, precision, recall,R1 and R5. Finally, as shown in the table and figures, we save the worst , good and best results.

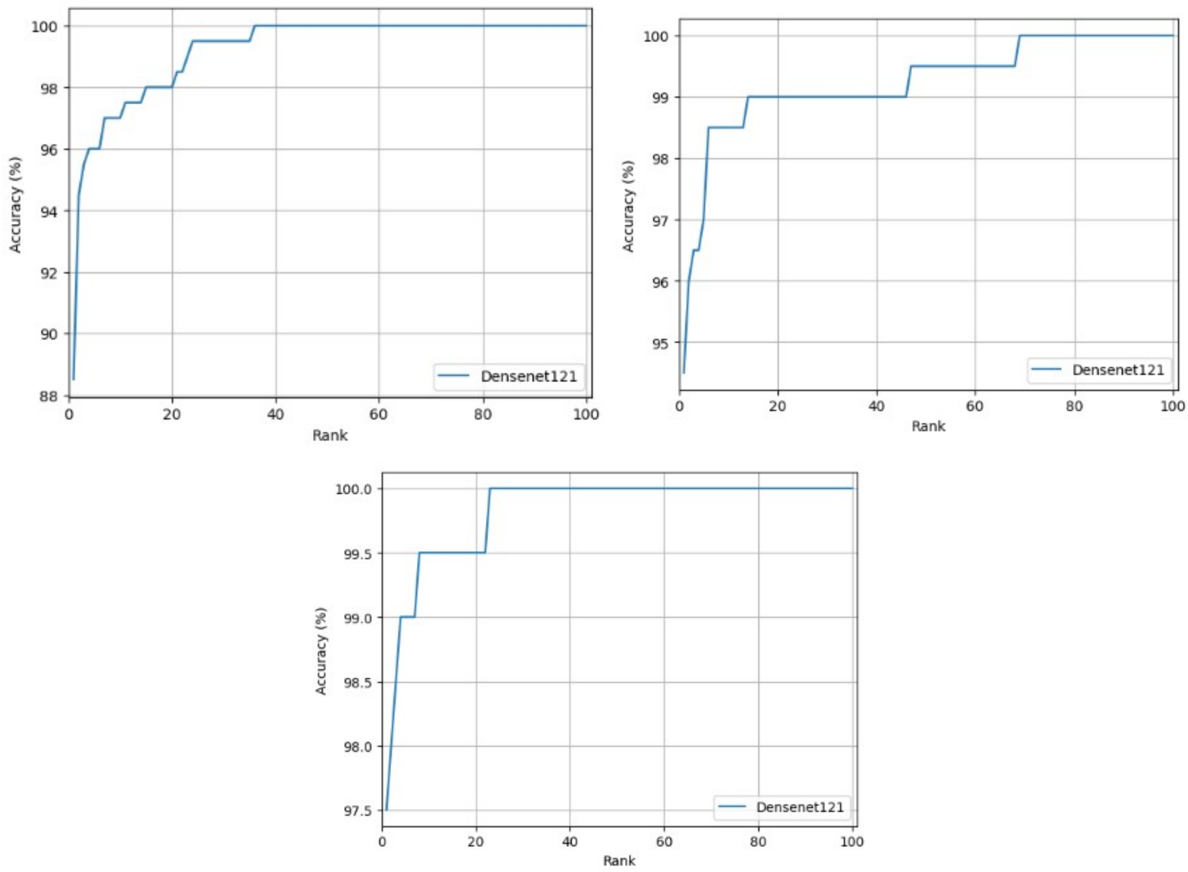

FIG 3.14 – *CMC curve using denseNet-121 on left dataset*

- Right Dataset
- The values selected are based on the previous tables Model denseNet-121 Right

In Table 3.24,we use the four parameters that are important for evaluating our model : number of epochs , batch size, learning rate and the optimizer. To begin, we select the best parameters from the previous table. The F1-score, precision, recall,R1 and R5. Finally, as shown in the table and figures, we save the worst , good and best results.

Metrics(%)	Learning rate= 0.01 Optimizer= RMSprop Batch size=32 Number of epochs= 50	Learning rate= 0.006 Optimizer= RMSprop Batch size=32 Number of epochs= 40	Learning rate= 0.006 Optimizer= RMSprop Batch size= 16 Number of epochs= 30
R1	63	91	94.00
R5	84	96	98.50
Precission	64	94	95
Recall	63	91	94
F1score	59	91	94

TABLE 3.24 – *Values selected DenseNet-121 Right*

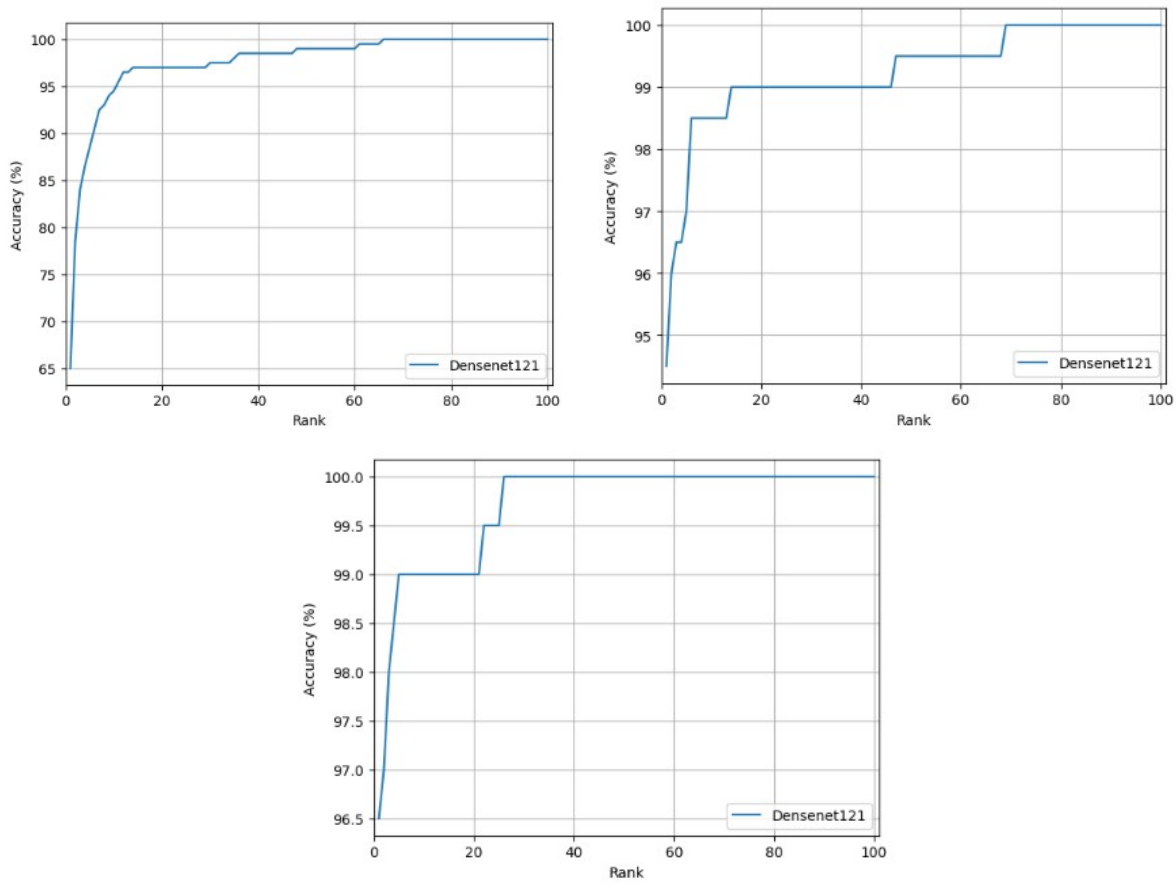

FIG 3.15 – *CMC curve using DenseNet-121 on Right dataset*

- Left+Right Dataset
- The values selected are based on the previous tables Model DenseNet-121 Left+Right

In Table 3.25, we use four parameters that are important for evaluating our model : number of epochs, batch size, learning rate, and optimizer. To begin, we select the best parameters from the previous table. We then evaluate the model using F1-score, precision, recall, R1, and R5. Finally, as shown in the table and figures, we present the worst, good, and best results.

Metrics(%)	Learning rate= 0.001 Optimizer= RMSprop Batch size=16 Number of epochs= 35	Learning rate= 0.005 Optimizer= RMSprop Batch size=32 Number of epochs= 50	Learning rate= 0.004 Optimizer= RMSprop Batch size= 64 Number of epochs= 35
R1	84	90	96.50
R5	93	96	99
Precission	91	94	97
Recall	83	90	96
F1score	85	91	96

TABLE 3.25 – *Values selected DenseNet-121 Left+Right*

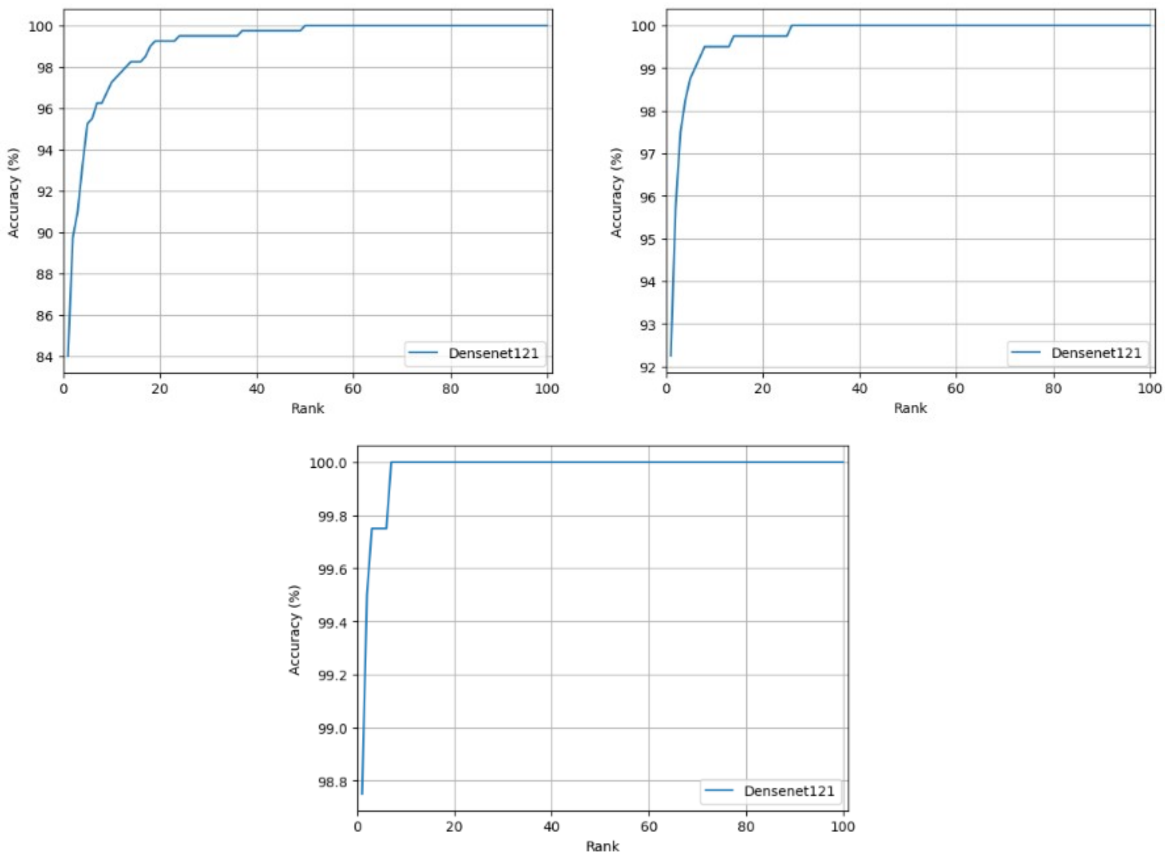

FIG 3.16 – *CMC curve using DenseNet-121 on Left+Right dataset*

4.2.3 MobileNet-V2 results

- Influence of optimizer and learning rate on the MobileNet-V2 performance using fine tuning technique

In Table 3.26, we set the number of epochs and batch size in advance and start changing learning rate and the optimizer,in data left until we get the worst value accuracy of 57% at learning rate 0.01and optimizer Adam ,the good value accuracy 75% at learning rate 0.002 and optimizer Adam and best estimated value accuracy of 90.05% at learning rate 0.006 and optimizer RMSprop, ,in data right until we get the worst value accuracy of 66% at learning rate 0.01 and optimizer RMSprop, the good value accuracy of 80% at learning rate 0.001and optimizer Adam and best estimated value accuracy of 91% at learning rate 0.006 and optimizer RMSprop. In

Datasets	Optimizer	Learning rate							
		0.01	0.001	0.002	0.0004	0.005	0.0005	0.006	0.0006
Left	Adam	57	60	75	83	80	85	83	80
	RMSprop	81	82.79	83	78	85	83.5	90.05	89.45
Right	Adam	67	80	81.50	81.50	85	83	83	80
	RMSprop	66	75	85.45	80	89	85	91	90
Left+Right	Adam	79.75	80	80	82	87	85	85	82
	RMSprop	83.50	85	85	89	91	90	92	90

TABLE 3.26 – *Results of Optimizers and Learning Rates for Different Datasets using MobileNet-V2*

multi-instance until we get the worst value accuracy of 79.75% at learning rate 0.01 and optimizer Adam , ,the good value accuracy of 85% at learning rate 0.0005 and optimizer Adam and best estimated value accuracy of 92% at learning rate 0.006 and optimizer RMSprop.

- Influence of epoch number and batch size on the MobileNet-V2 performance using fine tuning technique

Datasets	Epochs number	Batch size			
		16	32	64	128
Left	10	83.15	83	80.32	77
	25	90.45	89	85	80
	35	94	92.30	85	88
	40	95.50	94.05	90.80	90
	50	96.50	96	93	92
Right	10	90	88	89	88
	25	88	85	90	89
	35	90	88	88	89
	40	94	91	87	90
	50	93	91	91	90
Left+Right	10	89	90	91	91
	25	92	93	92	96
	35	97	95	90	94
	40	94	98	94	0
	50	98.50	95	96	96

TABLE 3.27 – *Results of Epoch Numbers and Batch Sizes for Different Datasets using MobileNet-V2*

In Table 3.27, we set the learning rate and optimizer to the best values from Table 3.26 and start varying the number of epochs and batch size. For the left data, we obtain the worst accuracy of 77% at 10 epochs and batch size 128, a good accuracy of 85% at 25 epochs and batch size 64, and the best estimated accuracy of 96.5% at 50 epochs and batch size 16. For the right data, we obtain the worst accuracy of 85% at 25 epochs and batch size 32, a good accuracy of 89% at 10 epochs and batch size 64, and the best estimated accuracy of 94% at 40 epochs and batch size 16. In multi-instance, we obtain the worst accuracy of 89% at 10 epochs and batch size 16, a good accuracy of 93% at 25 epochs and batch size 32, and the best estimated accuracy of 98.50% at 50 epochs and batch size 16.

- Left Dataset
- The values selected are based on the previous tables Model MobileNet-V2 Left

Metrics(%)	Learning rate= 0.01 Optimizer= Adam Batch size=32 Number of epochs= 50	Learning rate= 0.01 Optimizer= RMSprop Batch size=32 Number of epochs= 50	Learning rate= 0.006 Optimizer= RMSprop Batch size= 16 Number of epochs= 50
R1	57.50	81	96.50
R5	89.50	93.50	99
Precission	62	83	97
Recall	57	81	96
F1score	56	79	96

TABLE 3.28 – *Values selected MobileNet-V2 on Left*

In Table 3.28, we use the four parameters that are important for evaluating our model : number of epochs, batch size, learning rate, and optimizer. To begin, we select the best parameters from the previous table. We then evaluate the model using F1-score, precision, recall, R1, and R5. Finally, as shown in the table and figures, we present the worst, good, and best results.

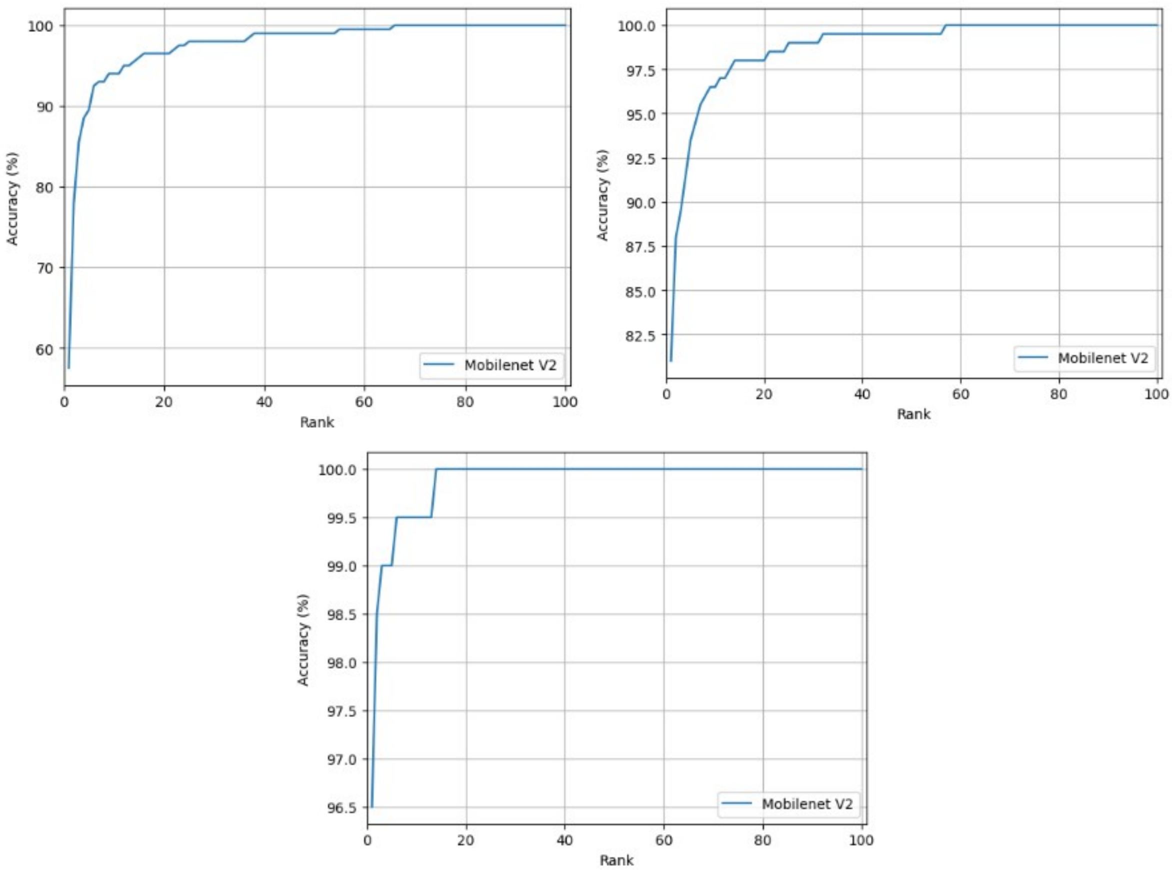

FIG 3.17 – *CMC curve using MobileNet-V2 on left dataset*

- Right Dataset
- The values selected are based on the previous tables Model MobileNet-V2 Right

Metrics(%)	Learning rate= 0.01 Optimizer= RMSprop Batch size=32 Number of epochs= 50	Learning rate= 0.006 Optimizer= RMSprop Batch size=32 Number of epochs= 40	Learning rate= 0.006 Optimizer= RMSprop Batch size= 16 Number of epochs= 30
R1	66	94.50	97.50
R5	92	98.50	98.50
Precission	68	96	98
Recall	66	94	97
F1score	63	94	98

TABLE 3.29 – *Values selected MobileNet-V2 Right*

In Table 3.29, we use the four parameters that are important for evaluating our model : number of epochs, batch size, learning rate, and optimizer. To begin, we select the best parameters from the previous table. We then evaluate the model using F1-score, precision, recall, R1, and R5. Finally, as shown in the table and figures, we present the worst, good, and best results.

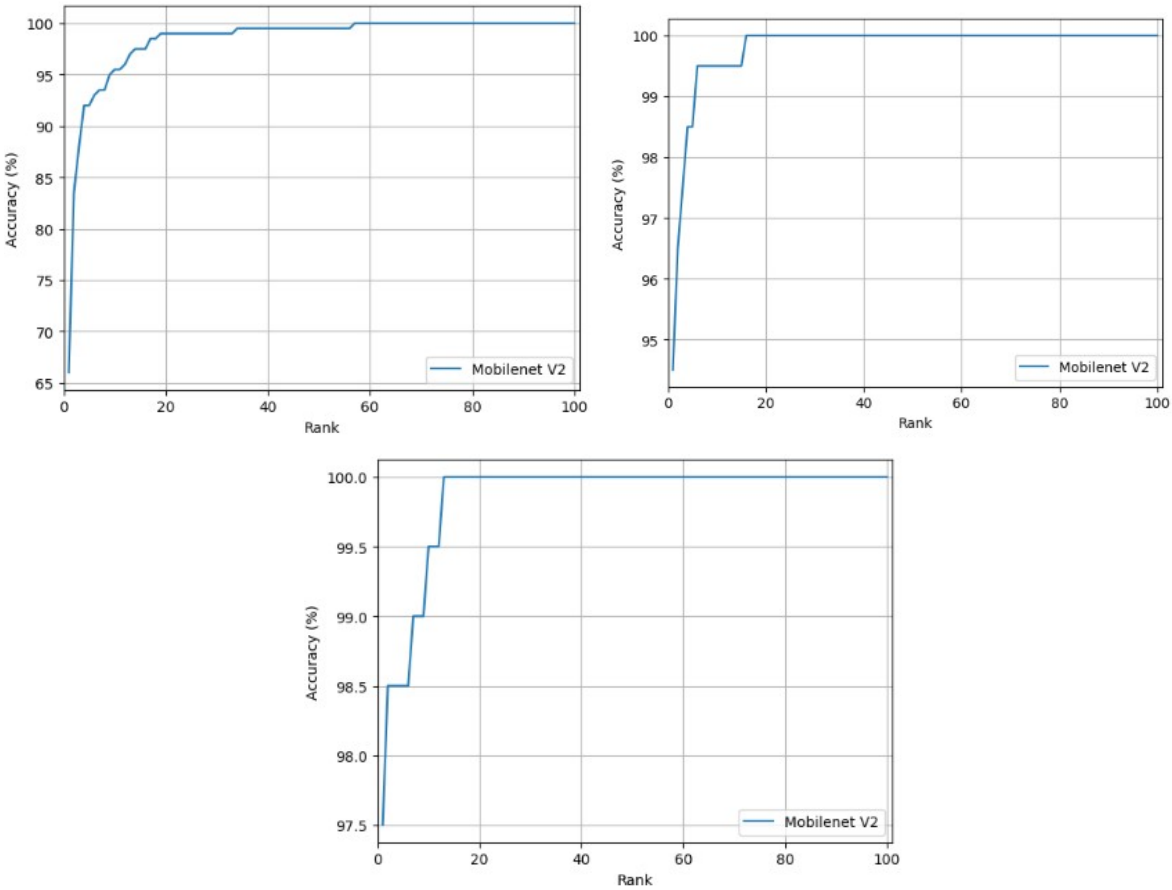

FIG 3.18 – *CMC curve using MobileNet-V2 on Right dataset*

- Left+Right Dataset
- The values selected are based on the previous tables Model MobileNet-V2 Left+Right

In Table 3.3, we use four parameters that are important for evaluating our model : number of epochs, batch size, learning rate, and optimizer. To begin, we select the best parameters from

Metrics(%)	Learning rate= 0.001 Optimizer= RMSprop Batch size=16 Number of epochs= 35	Learning rate= 0.005 Optimizer= RMSprop Batch size=32 Number of epochs= 50	Learning rate= 0.004 Optimizer= RMSprop Batch size= 64 Number of epochs= 35
R1	68	96.25	98.50
R5	92.75	99	99.75
Precission	75	97	99
Recall	68	96	98
F1score	65	96	98

TABLE 3.30 – *Values selected MobileNet-V2 Left+Right*

the previous table. We then evaluate the model using F1-score, precision, recall, R1, and R5. Finally, as shown in the table and figures, we present the worst, good, and best results.

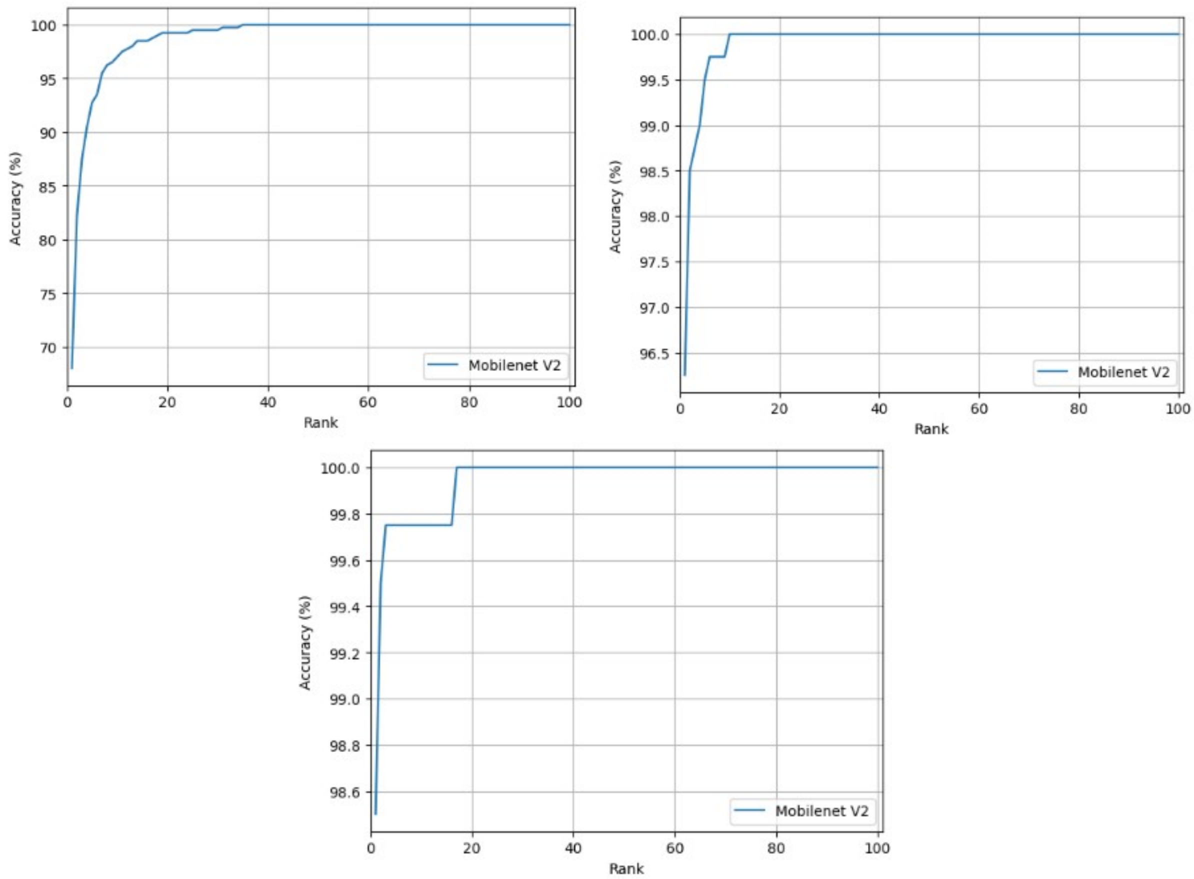

FIG 3.19 – *CMC curve using MobileNet-V2 on Left+Right dataset*

4.3 COMPARATIVE STUDY

In this section, we provide a meaningful comparison with other works that have utilized the same dataset for palmprint recognition. It is important to note that the number of subjects was 100. Table 3.31 presents the comparative statistics. From Table 3.31, we observe that our system performs exceptionally well with the fine-tuning experiment on the DenseNet-121 model compared to other methods. Despite variations in the number of subjects, our proposed system demonstrates high performance with an estimated accuracy of 98.75%, which is considered acceptable.

Reference	Year	Method	Dataset	Accuracy(%)
[Wan et al., 2023]	2023	LR-2DLDGE	Polyu	93.87
[Türk et al., 2023]	2023	CNN +SVM	Polyu	99.72
[Grosz et al., 2024]	2024	vit and cnn	Polyu	98.24
[Abdu-al Kadhm and Hasan,]	2024	IITD MERG	Polyu	95.50 99.93
our work	2024	DenseNET-121	Polyu	98.75%

TABLE 3.31 – *Comparison with state-of-the-art methods.*

5 Conclusion

After expriment transfer learning and expriment fine tuning with three different models in this chapter, it was DenseNet-121 that got the best result in accuracy compared to others models.

General Conclusion

Biometrics is a fascinating and intricate area. By employing sophisticated mathematical techniques, it makes an effort to differentiate between people, requiring us to operate within a highly diverse environment. The vast array of palmprint recognition algorithms that have been created also reflects this variety. By comparing and analyzing patterns based on palmprint features, the latter biometric software program may be used to uniquely identify or verify an individual. palmprint is an exceptional biometric in terms of confidentiality. readiness, ease of use, and safety. Compared to the face, the palmprint is more private since it is located on the inside of the hand and is thus almost hard to get without consent. Palmprint is more user-friendly since it has fewer usage constraints than iris and fingerprints. Additionally, palm veins and prints may be taken at the same time to create a dual-modal system that is extremely safe.

In this thesis, we focus on palmprint identification. Our primary objective is to develop a robust algorithm capable of recognizing individuals based on their unique palmprint patterns. To achieve this, we leverage the power of Convolutional Neural Networks (CNNs), one of the most recent techniques in this field for automatically extracting information.

Our research conclusively validates the effectiveness of our palmprint identification system. Its performance is remarkably robust, with the Densenet121 model significantly outperforming other methods. The system's exceptional accuracy rate of 98.75% firmly establishes its high reliability in biometric recognition. This approach has the potential to be compatible with over 100 subjects and be applied in many contexts to facilitate interpersonal relationships with little modifications. The system grows safer, the mistake rate goes down, and it becomes a significant future with extensive training and testing.

Bibliography

[Abdu-al Kadhm and Hasan,] Abdu-al Kadhm, S. J. and Hasan, T. M. Contactless palmprint recognition using deep learning technology.

[Albelwi, 2022] Albelwi, S. A. (2022). Deep architecture based on densenet-121 model for weather image recognition. *International Journal of Advanced Computer Science and Applications*, 13(10).

[Ali et al., 2016] Ali, M. M., Mahale, V. H., Yannawar, P., and Gaikwad, A. (2016). Overview of fingerprint recognition system. In *2016 international conference on electrical, electronics, and optimization techniques (ICEEOT)*, pages 1334–1338. IEEE.

[Aloysius and Geetha, 2017] Aloysius, N. and Geetha, M. (2017). A review on deep convolutional neural networks. In *2017 international conference on communication and signal processing (ICCSP)*, pages 0588–0592. IEEE.

[Alsaadi, 2015] Alsaadi, I. M. (2015). Physiological biometric authentication systems, advantages, disadvantages and future development : A review. *International Journal of Scientific & Technology Research*, 4(12) :285–289.

[Alsaadi, 2021] Alsaadi, I. M. (2021). Study on most popular behavioral biometrics, advantages, disadvantages and recent applications : A review. *Int. J. Sci. Technol. Res*, 10(1).

[Ammour et al., 2017] Ammour, B., Bouden, T., and Amira-Biad, S. (2017). Multimodal biometric identification system based on the face and iris. In *2017 5th International Conference on Electrical Engineering-Boumerdes (ICEE-B)*, pages 1–6. IEEE.

[Amrouni et al., 2022] Amrouni, N., Benzaoui, A., Bouaouina, R., Khaldi, Y., Adjabi, I., and Bouglimina, O. (2022). Contactless palmprint recognition using binarized statistical image features-based multiresolution analysis. *Sensors*, 22(24) :9814.

[Amrouni et al., 2023] Amrouni, N., Benzaoui, A., and Zeroual, A. (2023). Palmprint recognition : Extensive exploration of databases, methodologies, comparative assessment, and future directions. *Applied Sciences*, 14(1) :153.

[Ayodele, 2010] Ayodele, T. O. (2010). Types of machine learning algorithms. *New advances in machine learning*, 3(19-48) :5–1.

[Babich, 2012] Babich, A. (2012). Biometric authentication. types of biometric identifiers.

[Cao et al., 2012] Cao, X., Shen, W., Yu, L.-G., Wang, Y., Yang, J.-Y., and Zhang, Z. (2012). Illumination invariant extraction for face recognition using neighboring wavelet coefficients. *Pattern Recognition*, 45(4) :1299–1305.

[Chai et al., 2019] Chai, T., Prasad, S., and Wang, S. (2019). Boosting palmprint identification with gender information using deepnet. *Future Generation Computer Systems*, 99 :41–53.

[Dahea and Fadewar, 2018] Dahea, W. and Fadewar, H. (2018). Multimodal biometric system : A review. *International Journal of Research in Advanced Engineering and Technology*, 4(1) :25–31.

[Dargan and Kumar, 2020] Dargan, S. and Kumar, M. (2020). A comprehensive survey on the biometric recognition systems based on physiological and behavioral modalities. *Expert Systems with Applications*, 143 :113114.

[Dhameliya and Chaudhari, 2013] Dhameliya, M. D. and Chaudhari, J. P. (2013). A multimodal biometric recognition system based on fusion of palmprint and fingerprint. *International journal of Engineering trends and technology*, 4(5) :1908–1911.

[Dhanalakshmi and Srinivasan, 2013] Dhanalakshmi, A. and Srinivasan, B. (2013). Improved person identification system using face biometric detection. *International Journal of Advanced Networking and Applications*, 4(5) :1731.

[El-Abed et al., 2010] El-Abed, M., Giot, R., Hemery, B., and Rosenberger, C. (2010). A study of users' acceptance and satisfaction of biometric systems. In *44th Annual 2010 IEEE International Carnahan Conference on Security Technology*, pages 170–178. IEEE.

[El-Sayed, 2015] El-Sayed, A. (2015). Multi-biometric systems : a state of the art survey and research directions. *IJACSA) International Journal of Advanced Computer Science and Applications*, 6.

[El-Tarhouni et al., 2019] El-Tarhouni, W., Boubchir, L., Elbendak, M., and Bouridane, A. (2019). Multispectral palmprint recognition using pascal coefficients-based lbp and phog descriptors with random sampling. *Neural Computing and Applications*, 31 :593–603.

[Galbally et al., 2017] Galbally, J., Gomez-Barrero, M., and Ross, A. (2017). Accuracy evaluation of handwritten signature verification : Rethinking the random-skilled forgeries dichotomy. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 302–310. IEEE.

[Gandhe and Jawale, 2016] Gandhe, S. and Jawale, T. (2016). Human identilication using fusion of iris, signature and gait recognition. In *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, pages 282–285. IEEE.

[Genovese et al., 2019] Genovese, A., Piuri, V., Plataniotis, K. N., and Scotti, F. (2019). Palmnet : Gabor-pca convolutional networks for touchless palmprint recognition. *IEEE Transactions on Information Forensics and Security*, 14(12) :3160–3174.

[Grosz et al., 2024] Grosz, S. A., Godbole, A., and Jain, A. K. (2024). Mobile contactless palmprint recognition : Use of multiscale, multimodel embeddings. *arXiv preprint arXiv :2401.08111*.

[Gumaei et al., 2018] Gumaei, A., Sammouda, R., Al-Salman, A. M., and Alsanad, A. (2018). An effective palmprint recognition approach for visible and multispectral sensor images. *Sensors*, 18(5) :1575.

[Harakannanavar et al., 2019] Harakannanavar, S. S., Renukamurthy, P. C., and Raja, K. B. (2019). Comprehensive study of biometric authentication systems, challenges and future trends. *International Journal of Advanced Networking and Applications*, 10(4) :3958–3968.

[Hezil and Boukrouche, 2017] Hezil, N. and Boukrouche, A. (2017). Multimodal biometric recognition using human ear and palmprint. *IET Biometrics*, 6(5) :351–359.

[Izadpanahkakhk et al., 2018] Izadpanahkakhk, M., Razavi, S. M., Taghipour-Gorjikolaie, M., Zahiri, S. H., and Uncini, A. (2018). Deep region of interest and feature extraction models for palmprint verification using convolutional neural networks transfer learning. *Applied Sciences*, 8(7) :1210.

[Jain et al., 2007] Jain, A. K., Flynn, P., and Ross, A. A. (2007). *Handbook of biometrics*. Springer Science & Business Media.

[Kaur et al., 2014] Kaur, G., Singh, G., and Kumar, V. (2014). A review on biometric recognition. *International Journal of Bio-Science and Bio-Technology*, 6(4) :69–76.

[Khan et al., 2020] Khan, A., Sohail, A., Zahoora, U., and Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial intelligence review*, 53 :5455–5516.

[Kriegeskorte and Golan, 2019] Kriegeskorte, N. and Golan, T. (2019). Neural network models and deep learning. *Current Biology*, 29(7) :R231–R236.

[Li et al., 2022] Li, P., Jing, R., and Shi, X. (2022). Apple disease recognition based on convolutional neural networks with modified softmax. *Frontiers in plant science*, 13 :820146.

[Liu and Kumar, 2020] Liu, Y. and Kumar, A. (2020). Contactless palmprint identification using deeply learned residual features. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2(2) :172–181.

[Mahesh, 2020] Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, 9(1) :381–386.

[Matkowski et al., 2019] Matkowski, W. M., Chai, T., and Kong, A. W. K. (2019). Palmprint recognition in uncontrolled and uncooperative environment. *IEEE Transactions on Information Forensics and Security*, 15 :1601–1615.

[McCarthy et al., 2007] McCarthy, J. et al. (2007). What is artificial intelligence.

[Michele et al., 2019] Michele, A., Colin, V., and Santika, D. D. (2019). Mobilenet convolutional neural networks and support vector machines for palmprint recognition. *Procedia Computer Science*, 157 :110–117.

[Mittal and Garg, 2014] Mittal, M. and Garg, B. (2014). Secure identity using multimodal biometrics. *Int. J. Inf. Technol. Knowl*, 7(2) :20–25.

[Mou et al., 2017] Mou, L., Ghamisi, P., and Zhu, X. X. (2017). Deep recurrent neural networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7) :3639–3655.

[Murphy, 2016] Murphy, J. (2016). An overview of convolutional neural network architectures for deep learning. *Microway Inc*, pages 1–22.

[Oloyede and Hancke, 2016] Oloyede, M. O. and Hancke, G. P. (2016). Unimodal and multimodal biometric sensing systems : a review. *IEEE access*, 4 :7532–7555.

[Rida et al., 2018] Rida, I., Al-Maadeed, S., Mahmood, A., Bouridane, A., and Bakshi, S. (2018). Palmprint identification using an ensemble of sparse representations. *IEEE Access*, 6 :3241–3248.

[Ross and Jain, 2004] Ross, A. and Jain, A. K. (2004). Multimodal biometrics : An overview. In *2004 12th European signal processing conference*, pages 1221–1224. IEEE.

[Sabhanayagam et al., 2018] Sabhanayagam, T., Venkatesan, V. P., and Senthamaraikannan, K. (2018). A comprehensive survey on various biometric systems. *International Journal of Applied Engineering Research*, 13(5) :2276–2297.

[Shen et al., 2022] Shen, L., Zhang, Y., Zhao, K., Zhang, R., and Shen, W. (2022). Distribution alignment for cross-device palmprint recognition. *Pattern Recognition*, 132 :108942.

[Sultan et al., 2019] Sultan, H. H., Salem, N. M., and Al-Atabany, W. (2019). Multi-classification of brain tumor images using deep neural network. *IEEE access*, 7 :69215–69225.

[Tamrakar and Khanna, 2016] Tamrakar, D. and Khanna, P. (2016). Kernel discriminant analysis of block-wise gaussian derivative phase pattern histogram for palmprint recognition. *Journal of Visual Communication and Image Representation*, 40 :432–448.

[Thomas et al., 2016] Thomas, T., George, A., and Devi, K. I. (2016). Effective iris recognition system. *Procedia Technology*, 25 :464–472.

[Tiwari et al., 2015] Tiwari, S., Chourasia, J., and Chourasia, V. S. (2015). A review of advancements in biometric systems. *International Journal of Innovative Research in Advanced Engineering*, 2(1) :187–204.

[Torrey and Shavlik, 2010] Torrey, L. and Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends : algorithms, methods, and techniques*, pages 242–264. IGI global.

[Türk et al., 2023] Türk, Ö., Çalışkan, A., Acar, E., and Ergen, B. (2023). Palmprint recognition system based on deep region of interest features with the aid of hybrid approach. *Signal, Image and Video Processing*, 17(7) :3837–3845.

[Vrbančič and Podgorelec, 2020] Vrbančič, G. and Podgorelec, V. (2020). Transfer learning with adaptive fine-tuning. *IEEE Access*, 8 :196197–196211.

[Vyas et al., 2022] Vyas, R., Kanumuri, T., Sheoran, G., and Dubey, P. (2022). Accurate feature extraction for multimodal biometrics combining iris and palmprint. *Journal of Ambient Intelligence and Humanized Computing*, 13(12) :5581–5589.

[Wan et al., 2023] Wan, M., Chen, X., Zhan, T., Yang, G., Tan, H., and Zheng, H. (2023). Low-rank 2d local discriminant graph embedding for robust image feature extraction. *Pattern Recognition*, 133 :109034.

[Younis et al., 2022] Younis, A., Qiang, L., Nyatega, C. O., Adamu, M. J., and Kawuwa, H. B. (2022). Brain tumor analysis using deep learning and vgg-16 ensembling learning approaches. *Applied Sciences*, 12(14) :7282.

[Zhang et al., 2019] Zhang, J., Yu, P. S., Zhang, J., and Yu, P. S. (2019). Broad learning introduction. *Broad Learning Through Fusions : An Application on Social Networks*, pages 3–17.

[Zhao et al., 2022] Zhao, S., Wu, J., Fei, L., Zhang, B., and Zhao, P. (2022). Double-cohesion learning based multiview and discriminant palmprint recognition. *Information Fusion*, 83 :96–109.

[Zhong et al., 2019] Zhong, D., Du, X., and Zhong, K. (2019). Decade progress of palmprint recognition : A brief survey. *Neurocomputing*, 328 :16–28.

Abstract

These days, there is more talk about increasing crime, piracy, and lack of security across different sectors. It is also very important to verify people's identities for financial transactions, accessing services, and mobility. Traditional security systems use pre-existing information (like passwords orPINs) or token-based access (like keys, IDs, or badges). However, these systems frequently cannotdiscriminate between fraudsters and those who are allowed, they are less trustworthy in many environments. In this work, we choose to investigate one of these systems, which is a deep-learning palmprint recognition system.This system is difficult to replicate. There are several benefits, such as affordability and simplicity of usage. Our work may be categorized into two parts for feature extraction: transfer learning and fine-tuning, and two strategies: learning one instance and multiple instances. Firstly, we prepare our datasets into 3 datasets to evaluate our proposed models: left, right, and multiple instances. After that, we select four convolutional neural network algorithms to carry out the feature extraction and classification operation to confirm individual Recognition using both techniques: transfer learning and fine-tuning. The PolyU palmprint database is used to evaluate the performance of the suggested model. Our proposed method for the PolyU palmprint database using transfer learning achieved an accuracy of 85.25\% with VGG16, 87\% with DenseNet121, and 86.25\% with MobileNetV2. Using fine-tuning, we achieved an accuracy of 92.50\% with VGG16, 98.75\% with DenseNet121, and 98.50\% with MobileNetV2. Experimental results conclude that the proposed work obtained goodperformance compared to existing methods in multi-instance scenarios.
{Keywords}: Palmprint, Recognition, CNN, Feature extraction, multi-instance , one-instance.

Résumé

Ces jours-ci, il y a de plus en plus de discussions sur l'augmentation de la criminalité, de la piraterie etdu manque de sécurité dans différents secteurs. Il est également très important de vérifier l'identité des personnes pour les transactions financières, l'accès aux services et la mobilité. Les systèmes de sécurité traditionnels utilisent des informations préexistantes (comme des mots de passe ou des codes PIN) ou un accès basé sur des jetons (comme des clés, des identifiants ou des badges). Cependant, ces systèmes ne peuvent souvent pas distinguer les fraudeurs des utilisateurs autorisés, cequi les rend moins fiables dans de nombreux environnements. Dans ce travail, nous choisissons d'étudier l'un de ces systèmes, qui est un système de reconnaissance des empreintes de paume basé sur l'apprentissage profond. Ce système est difficile à reproduire. Il présente plusieurs avantages, tels que l'abordabilité et la simplicité d'utilisation. Notre travail peut être catégorisé en deux parties pour l'extraction de caractéristiques : le transfert d'apprentissage et le réglage fin, et deux stratégies : l'apprentissage d'une instance et de plusieurs instances. Tout d'abord, nous préparons nos ensembles de données en 3 ensembles pour évaluer nos modèles proposés : gauche, droite et plusieurs instances. Ensuite, nous sélectionnons quatre algorithmes de réseau neuronal convolutif pour effectuer l'extraction de caractéristiques et l'opération de classification afin de confirmer la reconnaissance individuelle en utilisant les deux techniques : transfert d'apprentissage et réglage fin. La base de données d'empreintes de paume PolyU est utilisée pour évaluer la performance du modèlesuggéré. Notre méthode proposée pour la base de données d'empreintes de paume PolyU utilisant le transfert d'apprentissage a atteint une précision de 85,25% avec VGG16, 87% avec DenseNet121, et 86,25% avec MobileNetV2. En utilisant le réglage fin, nous avons atteint une précision de 92,50% avec VGG16, 98,75% avec DenseNet121, et 98,50% avec MobileNetV2. Les résultats expérimentaux concluent que le travail proposé a obtenu de bonnes performances par rapport aux méthodesexistantes dans les scénarios multi-instance.

Mots-clés : Empreinte de paume Reconnaissance, CNN, Extraction de caractéristiques, multi-instance, one-instance

الملخص

في الآونة الأخيرة، تزايد الحديث حول ارتفاع معدل الجريمة والقرصنة وانعدام الأمن في مختلف القطاعات. كما أصبح من المهم للغاية التحقق من هوية الأشخاص لإجراء المعاملات المالية والوصول إلى الخدمات والتنقل. تعتمد أنظمة الأمان التقليدية على المعلومات المسبقة مثل كلمات المرور أو الأرقام السرية، أو طرق الوصول المعتمدة على الرموز مثل المفاتيح أو بطاقات الهوية أو الشارات. ومع ذلك، فإن هذه الأنظمة غالبًا ما تفشل في التمييز بين المحتالين والمستخدمين المصرح لهم، مما يجعلها أقل موثوقية في العديد من البيئات. في هذا البحث، نقوم بالتحقيق في نظام التعرف على بصمة الكف باستخدام التعلم العميق، وهو نظام يصعب تكراره. يقدم هذا النظام العديد من الفوائد، مثل التكلفة الميسورة وسهولة الاستخدام. يمكن تقسيم عملنا إلى جزأين لاستخراج الميزات: التعلم بالنقل والتدقيق الدقيق، واستراتيجيتين: التعلم من حالة واحدة ومن حالات متعددة. قمنا أولاً بتحضير مجموعات بياناتنا إلى ثلاث مجموعات لتقييم نماذجنا المقترحة: بصمة الكف اليسرى، وبصمة الكف اليمنى، والحالات المتعددة. بعد ذلك، اخترنا أربعة خوارزميات لشبكات الأعصاب التلافيفية لتنفيذ عملية استخراج الميزات والتصنيف لتأكيد التعرف على الأفراد باستخدام كلتا التقنيتين: التعلم بالنقل والتدقيق الدقيق. تم استخدام قاعدة بيانات بصمة الكف PolyU لتقييم أداء النموذج المقترح. باستخدام التعلم بالنقل على قاعدة بيانات بصمة الكف PolyU، حققت طريقتنا دقة تبلغ 85.25% مع VGG16، و87% مع DenseNet121، و86.25% مع MobileNetV2. باستخدام التدقيق الدقيق، حققنا دقة تبلغ 92.50% مع VGG16، و98.75% مع DenseNet121، و98.50% مع MobileNetV2. تُظهر النتائج التجريبية أن الطريقة المقترحة تتفوق على الطرق الحالية في سيناريوهات الحالات المتعددة.
الكلمات المفتاحية: بصمة الكف، التعرف، الشبكات العصبية التلافيفية، استخراج الميزات، الحالات المتعددة، الحالة الواحدة.