N°d'ordre :.....
N°de série :.....

People's Democratic Republic of Algeria

Ministry of Higher Education
and Scientific Research

ECHAHID HAMMA LAKHDAR UNIVERSITY- EL OUED

FACULTY OF EXACT SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

End of Study Thesis
Presented for the Diploma of

ACADEMIC MASTER

Domaine: Mathematics and Computer Science
Specialty : **Distributed Systems and Artificial Intelligence**

Theme

Development of human foot trace recognition system

Presented by:

AGGAB Sahar
REZZAG LOBZA Nadjah

In front of the committee composed of:

M. BOUCHERIT Ammar	MCA	President
Ms. KHOLLADI Nedjoua	MAA	Examiner
M. ZAIZ Faouzi	MAA	Supervisor

Academic year 2019/2020

Acknowledge

Thanks to ***ALLAH***
Almighty who gave us the strength
to complete this work.
Then a special thanks of gratitude to our supervisor
Dr. ZAIZ Faouzi
for his wonderful advice,
and to follow up our work.
also thanks for who helped us
in gathering different information,
collecting data during making this project.

Abstract

In the past, tracking science was known as the science of '*Al-Qiyafah*' where an expert nomad detects the important information about a person from its foot traces in the desert. Also, person identification is one of the most popular filed in computer vision. Nowadays, among numerous methods being developed for this purpose, biometric identification of finger or iris, or footprint scan is the most promising method. The process of person recognition using foot traces is much more challenging than using footprints because it does not have biometric features.

In this work, we aim to automate the process of 'Al-Qeiyafah' into an intelligent system. From an input image, the system extracts foot trace features using image processing techniques. The recognition system of traces uses advanced deep learning methods that helps to detect the traces and to recognize efficiently the features of each trace. In the proposed system, we use the convolution neural networks algorithm to predict the foot trace owner by referring to its features extracted from the performed system.

For this objective, we proposed a comparative study conducted to examine two different architectures of human foot trace recognition, the first one is based on one model approach. The second one is based on a multi-models approach with hierarchical taxonomy in order to gain processing time and to attain better accuracy. In both approaches, the classification system combines two models, a pre-trained model as a features extractor, and a classifier model. The pre-trained Xception model was nominated from five pre-trained models for integration into system classifiers. Besides, we examined this nominated pretrained model (Xception) with 3 types of classifiers, Softmax, SVM, and KNN.

The obtained results show clearly their accuracy, the first approach accuracy got 76.40% with an SVM classifier. while the accuracy of the second approach is the average accuracy of its models, and it got a significant result of 90.46% with a Softmax classifier. Therefore, we conclude that the second system works better than the first one but it is still insufficient to solve the aspect of human foot traces recognition.

Keywords: Image processing, Convolutional Neural Network, SVM, KNN, Softmax layer, Xception, Pre-trained models, multi-models, Foot trace, Tracking.

Résumé

Dans le passé, la science du suivi était connue comme la science de «*Al-Qiyafah* où un expert nomad détecte les informations importantes sur une personne à partir de ses traces de pied dans le désert. En outre, l'identification des personnes est l'un des domaines les plus populaires de la vision par ordinateur. De nos jours, parmi les nombreuses méthodes développées à cet effet, l'identification biométrique telle qu'un doigt ou un iris, ou encore le scan d'empreintes est la méthode la plus prometteuse actuellement. le processus de reconnaissance des personnes à l'aide de traces de pieds est beaucoup plus compliquée que l'utilisation d'empreintes car elle ne fournit pas de traits biométriques.

Dans ce travail, nous visons à automatiser le processus d '«*Al-Qeiyafah*» dans un système intelligent. À partir d'une image d'entrée, le système extrait les caractéristiques de trace de pied à l'aide de techniques de traitement d'image. Le système de reconnaissance des traces utilise des méthodes avancées d'apprentissage en profondeur qui aide à détecter les traces et à reconnaître efficacement les caractéristiques de chaque trace. dans le système proposé, nous utilisons l'algorithme des réseaux de neurones à convolution pour prédire le propriétaire de la trace du pied en se référant à ses caractéristiques extraites du système effectué.

Pour cet objectif, nous avons proposé une étude comparative menée pour examiner deux expériences différentes de reconnaissance de trace de pied humain, la première étant basée sur une approche modèle. Le second est basé sur une approche multi-modèles avec taxonomie hiérarchique afin de gagner en temps de traitement et une meilleure précision. Dans les deux approches, le système de classification combine deux modèles, un modèle pré-formé comme extracteur de caractéristiques et un modèle de classification. Le modèle Xception pré-formé a été nommé à partir de cinq modèles pré-formés pour l'intégration dans les classificateurs système. En outre, nous avons examiné ce modèle pré-entraîné désigné (Xception) avec 3 types de classificateurs, Softmax, SVM et KNN.

Les résultats des expériences appliquées montrent clairement leur précision, que la première précision d'approche obtenue (76,40%) avec un classificateur SVM. tandis que la précision de la deuxième approche est la précision moyenne de ses modèles, et elle a obtenu un résultat significatif (90,46%) avec un classificateur Softmax. Par conséquent, on en conclut que le second système fonctionne mieux que le premier mais il est encore insuffisant pour résoudre l'aspect de la reconnaissance des traces de pied humain.

Mots Clés: Traitement d'image, Réseaux de neurones convolutionnels, machine à super vecteur, k-voisins les plus proches, couche Softmax, Xception, Modèles pré-formés, Multimodèles, Trace de pied, Suivi.

الملخص

قديما كان علم التتبع يعرف بعلم 'القيافة' حيث يكتشف الخبير البدوي معلومات مهمة عن الانسان من خلال آثار أقدامه في الصحراء. أيضا، يعد التعرف على الاشخاص أحد أكثر الاغراض شيوعا في مجال رؤية الكمبيوتر. في الوقت الحاضر، من بين العديد من الأساليب التي يتم تطويرها لهذا الغرض، لدينا التعرف على القياسات الحيوية مثل بصمة الإصبع أو القزحية أو مسح بصمة القدم هي الطريقة الواعدة الآن. و بالمثل، يعد التعرف على الأشخاص باستخدام آثار الأقدام أكثر تعقيدا من استخدام مسح بصمة الاقدام لأنه لا يوفر ميزات بيومترية.

في هذا العمل، نهدف إلى جعل عملية 'القيافة' آليا في نظام ذكي. من صورة الادخال، يستخرج النظام ميزات أثر القدم باستخدام تقنيات معالجة الصور. يستخدم نظام التعرف على الآثار أساليب التعلم العميق المتقدمة التي تساعد على اكتشاف والتعرف بكفاءة على ميزات كل أثر. في النظام المقترح، نستخدم خوارزمية الشبكات العصبية الالتفافية للتنبؤ بمالك أثر القدم من خلال الرجوع إلى ميزاته المستخرجة من النظام المنفذ.

لهذا الغرض، اقترحنا دراسة مقارنة أجريت لفحص تجربتين مختلفتين للتعرف على أثر القدم البشرية، يعتمد الأول على نهج نموذجي واحد. و يعتمدالثاني على نهج متعدد النماذج مع تصنيف هرمي من أجل الحصول على وقت قياسي للمعالجة وتحقيق دقة أفضل. في كلا المنهجين، يجمع نظام التصنيف بين نموذجين، نموذج مدرب مسبقا كمستخرج ميزات، ونموذج مصنف. تم ترشيح نموذج اكسبشن المدرب مسبقا من خمسة نماذج مدربة مسبقا للتكامل مع مصنفات النظام. الى جانب ذلك، قمنا بفحص هذا النموذج المعين مسبقا (اكسبشن) مع ٣ أنواع من المصنفات سوفت ماكس، آلة ناقلات سوبر، و ك-أقرب الجيران.

تظهر النتائج المتحصل عليها بوضوح دقتها، حيث حصلت دقة النهج الأول على ٠٤.٦٧٪ مع مصنف آلة ناقلات سوبر. بينما دقة الطريقة الثانية هي متوسط دقة نماذجها و حصلت على نتيجة ملحوظة بنسبة ٪ ٦٤.٠٩ مع مصنف سوفت ماكس. وبالتالي، نستنتج أن النظام الثاني يعمل بشكل افضل من الأول و لكنه لا يزال غير كاف لحل كل جوانب التعرف على أثر اقدام الانسان.

كلمات مفتاحية: معالجة الصورة، الشبكات العصبية التلافيفة، آلة ناقلات سوبر، ك-أقرب الجيران، طبقة سوفت ماكس، اكسبشن، نماذج مدربة مسبقا، متعدد النماذج، أثر القدم، التتبع.

Table of contents

Acknowledge **i**

Abstract **ii**

Table of contents **vi**

List of figures **ix**

List of tables **xii**

Liste of algorithms **xiii**

Acronyms **xiv**

Introduction **1**

1 State of the art **3**

Introduction 3

1.1 basic concepts 3

1.1.1 Foot 3

1.1.2 Footwear 4

1.1.3 Trace 4

1.2 Foot trace recognition 5

1.3 Technical disciplines 6

1.3.1 Machine vision 7

1.3.2 Computer Vision 7

1.3.3 Image Recognition 9

1.4 Technical methods 9
1.4.1 Machine learning 9
1.4.2 Deep Learning 13
1.5 Comparison of deep learning and machine learning 19
Conclusion 19

2 Human identification from foot marks **20**
Introduction 20
2.1 Barefoot prints comparison methods 20
2.1.1 The Gunn Method 21
2.1.2 The Optical Center Approach 22
2.1.3 The Overlay Method 23
2.1.4 The Robbins Method 24
2.1.5 The Rossi's Podometric System 24
2.2 Identification involving footwear 25
2.2.1 Outer impressions 26
2.2.2 Inner impressions 26
2.3 Forensic Gait Analysis 27
Conclusion 27

3 Design and implementation **28**
Introduction 28
3.1 Dataset generation 29
3.1.1 Data acquisition 29
3.1.2 Data preprocessing 31
3.1.3 Dataset constructing 39
3.1.4 Dataset preparing 46
3.2 Systems models construction 47
3.2.1 Feature extractor 48
3.2.2 Classification 52
3.2.3 Matching 55
3.2.4 Making a decision 59
Conclusion 59

4 Results and discussions **60**
Introduction 60
4.1 Implementation tools 60
4.2 Implementation results 63
4.2.1 Training phase 64
4.2.2 Testing phase 65
4.3 Results discussion 67
4.4 Application interface 68
4.4.1 One model system interface 70
4.4.2 Multi-models system interface 75
Conclusion 82

Conclusion and future work **83**

Bibliographie **84**

List of figures

1 State of the art **3**

1.1 Scan footprint at the crime scene [1]. 6
1.2 Topography analysis scans [1]. 6
1.3 Essential elements of a machine vision system [2]. 7
1.4 Typical steps in computer vision systems [3]. 8
1.5 Supervised learning [4]. 10
1.6 Simple multi-layer neural network structure [5]. 11
1.7 Architecture of a Support Vector Machine classifier [6]. 11
1.8 Illustration of the k-nearest neighbors (kNN) classifier [7]. 12
1.9 Unsupervised learning [4]. 12
1.10 Semi-supervised learning [8]. 13
1.11 Reinforcement Learning, Agent and Environment [9]. 13
1.12 Typical deep neural network [10]. 14
1.13 A network with many convolutional layers [5]. 14
1.14 VGG16, VGG19, Inception V3, Xception and ResNet-50 architectures [11]. 17

2 Human identification from foot marks **20**

2.1 The Gunn Method [12]. 21
2.2 Development of the Gunn method[13]. 22
2.3 The Optical Center Method [13]. 23
2.4 The Overlay Method [14]. 23
2.5 The Robbins Method[13]. 24
2.6 Rossi's system of podometrics[12]. 25
2.7 Focal point instrument[14]. 26
2.8 Focal point instrument [14]. 27

3 Design and implementation **28**

3.1 General HFTs design process. 29
3.2 Volunteers information. 30
3.3 Sample of collected raw data. 30
3.4 Sample of the selected (a) and excluded (b) images. 32
3.5 Foot traces not cleaned. 32
3.6 Sample of cleaned data. 33

3.7 Image RGB color of footwear trace and its histogram. 33
3.8 Image RGB color of foot steps trace and its histogram. 34
3.9 Image RGB color of barefoot trace and its histogram. 34
3.10 Grayscale of a balanced image with its histogram. 35
3.11 Grayscale of lighted image with its histogram. 35
3.12 Grayscale of darkened image with its histogram. 36
3.13 Color image-3 with its histogram. 36
3.14 Binary foot image and its histogram. 37
3.15 Binary foot steps image and its histogram. 37
3.16 image-1 Before and after applying the contrast stretching. 38
3.17 image-2 Before and after applying the contrast Stretching. 38
3.18 image-3 Before and after applying the contrast stretching. 39
3.19 Tree structure of dataset for OMS. 41
3.20 Tree structure of dataset for MMS before optimization. 43
3.21 Dataset classes structure for MMS after optimization 44
3.22 Char 1: Dataset human foot trace statistics. 46
3.23 General HFT system scheme. 48
3.24 HFT of One model system schema. 49
3.25 HFT of the multi-models system schema. 50
3.26 Features extractor in the Xception. 52
3.27 Model architecture. 53
3.28 Model case1 (Softmax). 54
3.29 Model case2 (SVM). 54
3.30 Model case3 (KNN). 55

4 Results and discussions **60**
4.1 Split training set and testing set of each dataset. 64
4.2 Split training set and validation set of each dataset. 64
4.3 Prediction results of different classifiers for MMS. 66
4.4 Home interface for web app. 68
4.5 Home interface for mobile app. 69
4.6 OMS home interface for web app. 70
4.7 OMS home interface for mobile app. 70
4.8 OMS prediction results for the web app. 71
4.9 OMS prediction results for the mobile app. 72
4.10 OMS details prediction results for the web app. 73
4.11 OMS details prediction results for the mobile app. 73
4.12 OMS information of suitable person for the web app. 74
4.13 OMS information of suitable person for the mobile app. 74
4.14 MMS home interface for web app. 75
4.15 MMS home interface for mobile app. 75
4.16 MMS prediction results for the web app. 76
4.17 MMS prediction results for the mobile app. 77
4.18 MMS details prediction results for the web app. 78
4.19 MMS details prediction results for the mobile app. 78
4.20 MMS details prediction of 5 top persons for the web app. 79

4.21 MMS details prediction of 5 top persons for the mobile app. 80
4.22 MMS details of matching score information for the web app. 81
4.23 MMS details of matching score information for the mobile app. 81

List of tables

1 State of the art **3**
1.1 Comparison of deep learning and machine learning [15]. 19

2 Human identification from foot marks **20**
2.1 Different aspects of barefoot prints comparaison methods. 25

3 Design and implementation **28**
3.1 Processed data statistics for 55 individuals. 40
3.2 All dataset with its total size. 45
3.3 Results of a comparative experiments of different feature extractor models. 51

4 Results and discussions **60**
4.1 Splitting dataset to training set, testing set and validation set. 63
4.2 Results of testing the trained model (s) OMS & MMS. 66
4.3 Summarized results of testing OMS & MMS. 67

List of Algorithms

3 Design and implementation **28**
1 Matching in OMS 56
2 Matching in MMS 58

Acronyms

HFT: Human Foot Trace.
HFTs: Human Foot Trace system.
MMS: Multi Model System.
OMS: One Model System.
MV: Machine Vision.
CV: Computer Vision.
IP: Image Processing.
IR: Image Recognition.
AI: Artificial Intelligence.
ML: Machine Learning.
DL: Deep Learning.
NN: Neural Network.
CNN: Convolusion Neural Network .
SVM: support vector machine.
KNN: K-Nearest Neighbor.
S: Straight.
IT: In-Toeing.
OT: Out-Toeing.

Introduction

Recognition objects in visual data have been a popular area of research. Recently, image recognition is a well-studied field in computer vision [16], also, we see that deep learning has been widely used for image recognition [17]. Images recognition of foot traces are more difficult than images recognition of footprints, because the footprint image is so clear and proved to be distinct for every human being due to biometric features [18], unlike the other.

Foot trace recognition study is one of the newly emerging sciences which has many challenges in various aspects, such as the unavailability of the dataset is one of the biggest barriers in the success of this study [19]. The lack of scientific competencies and material capabilities that specialize in collecting accurately the traces. This lack leads to a loss of foot traces features. The whereabouts of the traces is an ever-changing space due to climatic factors. The climate change factors lead to hide or erase the traces.

In the criminal science field, many technical investigators are working on developing intelligent systems to recognize and identify the traces of the crime scene [20]. In addition, tracking science is very important for saving the lives of many beings especially humans. Also, the loss of people in the desert due to the difficult conditions in this region, and the lack of communication means, which makes the search in this area and delving into it very difficult. Depending on this phenomenon, fast technical solutions must be found to help find the missing people in the shortest time.

This work is about human foot trace recognition and identification in the desert based on developing a system that uses advanced computer vision techniques. The image data of foot trace can be collected by various means such as Drone [21], satellite planes, ...etc. The proposed system enables the integration of these means as an image input devices for the system.

Deep learning has very high efficiency and accuracy when it processes images by the convolution neural networks [19]. Using CNNs, we build our architecture and train it. Through this research, we present a comparative study of two different systems architectures. The first one is based on one model approach. While the second one was based on a multi-models approach which uses 9 hierarchical models structure. Both architectures uses CNNs model for extracting the images features. Each model is composed from two parts, pretrained model and classifier. The first part is a pre-trained

model as features extractor that was nominated from five pre-trained models, Xception, VGG16, VGG19, ResNet, and Inception V3. While the second part presents our proposed classifier which can be a softmax, SVM, or KNN classifier.

This thesis includes four chapters. In the first chapter, we present the most important concepts in this study, the different methods and techniques used to build the systems architectures. This chapter consists of five subheadings: the first section is about the basic concepts of the foot trace such as foot, footwear, and trace. While the second section is an overview of foot trace recognition between the past and the present. The third section presents the technical disciplines in a visual system which are: machine vision, computer vision, image recognition and image processing. In the fourth section, we show two technical methods: Machine learning and Deep learning, then we conclude with a comparison between these methods in the last section.

Then, in the second chapter, we define human identification methods from considering marks left by the foot on its surrounding environment. This chapter consists of three main subheadings: in the first section, we show several techniques of barefoot prints comparison methods and the anatomical measurements used to describe footmarks for each method. While the second section: Identification of involving footwear, we present tow possible ways to obtain information from shoes. In the last section, we present two possible ways of evaluation which are used in the forensic gait analysis.

The third chapter is the main chapter because it defines our proposed systems by explaining each system architecture (one model / multi-model), and the important general steps in the HFTs design process. we categorize these steps into two main subheadings: the dataset generation and the systems model (s) construction. In the first section, we generate and prepare data sets from primitive data to train our proposed models, and for the second section, we build the models through training and testing steps.

The last chapter is organized in four main subheadings, in the first section, we present the important development tools that we used in our work, in the second part, we implement the experiment on all models for each system, then in the third section we present and discuss the obtained results with testing different systems architectures. In the final section, we save and load the selected models in two different applications (web, mobile) and we show these applications interfaces how it works.

Finally, we conclude the work with a general conclusion concerning the obtained results of a comparative study of two different systems architectures with the used techniques and future works for improving the HFT system performance.

Chapter 1

State of the art

Introduction

The traces are important in forensic diagnoses or police criminal laboratories, as it is indicated by Edmond Locard, who created the first forensic laboratory in Lyon, stated a principle that is still valid today: "Every contact leaves a trace" [22, 23]. The traces help to know the truth of what happened before leaving the place of traces and who is its owner. Besides, It is an evidence of the existence of something, which helps to solve cases problems and discover mysterious incidents. Building a tracking recognition system facilitates police investigations and also speeds up diagnostics in forensic laboratories.

In this chapter, we will highlight the human foot trace in the desert. So, we review the important basic concepts about a foot trace. Also, we present the current approaches and methods used in this area of research.

1.1 basic concepts

In this section, we present the basic concepts about a foot trace such as foot, footwear, trace,. . . etc. Also, we indicate the the types of traces, i.e. the trace referred to a human or animals, or many things. Besides, the places categories where the traces can be found.

1.1.1 Foot

The foot is a complex and strong mechanical structure. It is an anatomical region that may demonstrate a high degree of individuality. As recognized by Defoe (1719), the human foot is also distinctive from the foot of non-humans. It is unique because it is best suited to the style of bipedal locomotion. The variations that scientists have found in foot bones for australopiths suggest there was variation in how they walked even among themselves [13].

1.1.2 Footwear

Footwear refers to garments worn on the feet, which originally serves the purpose of protection against adversities of the environment [24], the footwear has a peripheral association with an individual and is a body part, is what separates footwear marks from finger and other body part marks. Footwear is personal only to one owner, but it can also be owned by too many people.

1.1.3 Trace

Trace is a trial, sign, or vestige, or evidence of the former existence,. . . etc [25]. All of these meanings indicate that a mark left by something or someone was present and left [26]. The trace can be referred to a foot or hand, or many things.
Traces can be classified into three types:

- **Human**: like traces of humans (palm and feet, fingers, and body. . .). As for the foot trace, there are two types, with shoes (footwear) and without shoes (barefoot).
- **Animal**: like traces of animals (footprints of birds, reptiles, and insects. . .).
- **Things**: like traces of cars and bicycles, buses, and traces of pulling something.

Traces are found in different places that differ according to their physical nature, geographical location, and climate. Therefore, these places are classified according to their characteristics into three categories [27]:

- **Sand**: is a loose granular material that results from the disintegration of rocks, consists of particles smaller than gravel but coarser than silt [26]. Beach sand has been broken down predominantly by the action of water, whereas desert sand has been broken down by the wind. For each case, It should be noted that due to the nature of the sand, the trace in the desert sand is less visible than in the sandy beach. In addition to the rapid disappearance of monuments in the desert sand due to external factors such as wind.
- **Floor**: is the lower inside surface of a hollow structure (a ground surface) [26], the trace is left on the ground by specific materials and factors such as a dusty trace means that the raised part of the sole is coated with a layer of material (chalk, sawdust, dust . . .), or a colored trace results from the deposition of colored matter (blood, paint . . .) [27].
- **Snow**: is precipitation in the form of small white ice crystals formed directly from the water vapor of the air at a temperature of less than 32°F (0°C) [26]. Where there is a lot of snow, it leaves traces. It also disappears by the sun.

Note:
There's a difference between footprint and foot trace, the distinctions in the degree of appearance, The footprints appearance is quite predictable, and even the average person would recognize, with a reasonable degree of confidence, an area of friction-ridge detail associated with a footprint. Particularly, if the shape of the foot is apparent [13]. But the foot trace appearance is less clear than a footprint.

In this section, we define the recognition of the foot trace and its overview. Also, we define the basic technical disciplines related to the field of study: machine vision, computer vision, image recognition. In addition to the important methods used in the recognition area, machine learning, and deep learning.

1.2 Foot trace recognition

Foot trace recognition is the ability to track the foot trace, specifically the traces of animals and humans. It is called the science of Al-Qiyafah (the science of tracking the foot trace) in the Arabs. As the first appearance of this science with the human in primitive life for hunting [28]. It was also mentioned In the Holy Quran in Surah (chapter) Al-Kahf (The Cave) in the story of the prophet Moses and the Whale, [18.64] 'This is what we have been seeking! said he and they retraced their footsteps' [29].

Beginning in the middle of the 9th century before birth, Arabs were the most brilliant people in tracking the trace that called Al-Qiyafah. The science of Al-Qiyafah is a traditional human knowledge of tracing in the Arabian desert [30]. It helps to distinguish between a woman's foot and a man's foot, the young man's feet and the young woman's feet, and determining their height and weight, and more [28].

Until it became an accredited profession used to detect criminals, thieves, deserters, and missing persons. Most of the trackers are nomads and desert dwellers. Throughout history, they have played a very important role in criminal investigations. Many modern security services still rely on a tracker (the expert person who can identify and track a footprint) for criminal investigations. Now, forensic experts use the Artec MHT scanner in order to obtain a three-dimensional model of the trace [31, 28, 1], as shown in figure 1.1.

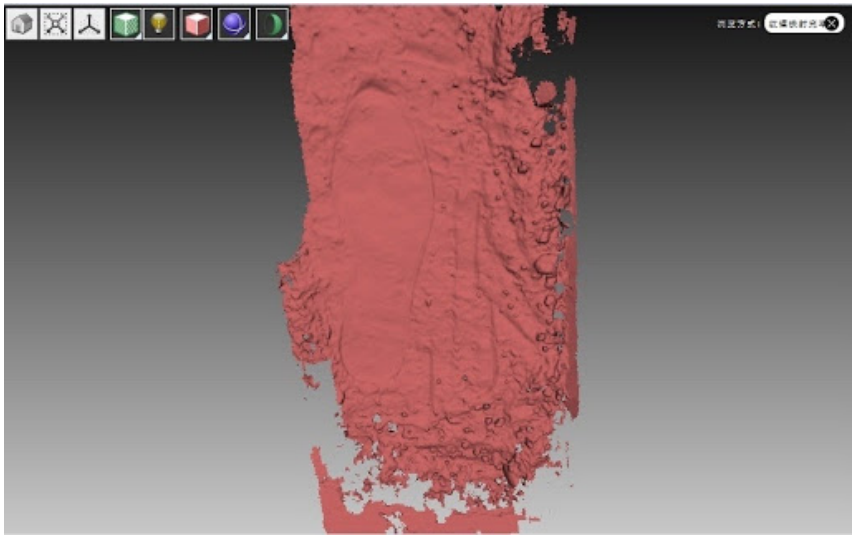


Figure 1.1: Scan footprint at the crime scene [1].

As technology develops, there is a lot of research in this field among them the special software developed by Beijing Coase Instrument Company Limited and Beijing University of Technology-CAUP, the Zhoukou police, it is not yet available as it is still being tested [1]. This software allocates different colors to different depths of the footprint topography as can be seen in figure 1.2.

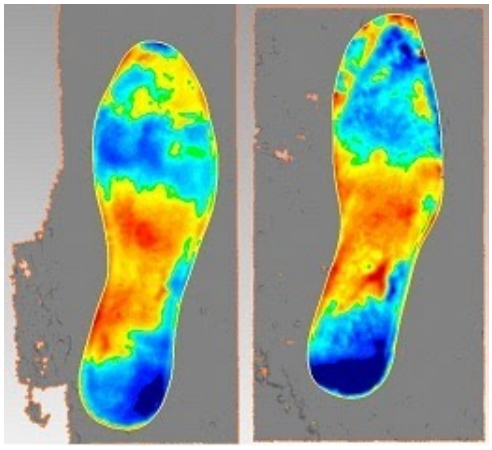


Figure 1.2: Topography analysis scans [1].

With the development of technology, there is a lot of research in this field but some of them are not yet available as it is still being tested [1].

1.3 Technical disciplines

In the visual system, the technical disciplines deal with searching the ways to automate all the jobs that a human visual system can do. Here, we present technical disciplines which are machine vision, computer vision, image recognition, and image processing. In addition, we present the global architecture of a computer vision system.

1.3.1 Machine vision

Machine vision is the ability of a computer to see anything like a human being. It is the use of devices for optical non-contact sensing to automatically receive and interpret an image of a real scene, in order to obtain information and/or control machines, or processes the information[32]. Figure 1.3 represents the important elements of a machine vision system.

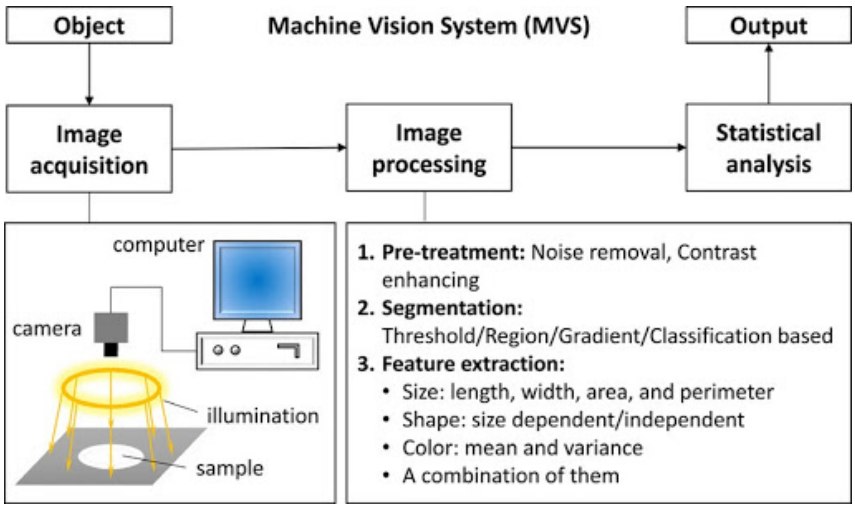


Figure 1.3: Essential elements of a machine vision system [2].

Machine vision employs one or more video cameras, analog-to-digital conversion (ADC), and digital signal processing (DSP). The resulting data goes to a computer or robot controller [33, 34, 35]. Machine vision includes methods for acquiring, processing, analyzing, and understanding images and, in general, high-dimensional data from the real world in order to produce numerical or symbolic information, e.g., in the forms of decisions.

1.3.2 Computer Vision

Computer vision is the transformation of data from a still or video camera into either a decision or a new representation [36]. Within computer science, that deals with enabling computers, machines, or devices. In general, to see, understand, interpret, or manipulate what is being seen [37]. The general goal of computer vision is to derive information about the scene by computer analysis of images of that scene [38].

1.3.2.1 Computer vision technique

Computer vision as a system uses algorithms of many techniques to perform the emulation of vision at the human scale or enhance the images. Among them, image processing.

Image processing is considered one of the emerging techniques. It is a technical method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it [39]. Image processing is a type of signal processing in which input is in image form and output

can be characteristics/features associated with an image. Processing steps include image acquisition tools, analysis/manipulation, and the results in the form of image or report that is based on image analysis[30].

1.3.2.2 Computer vision system

Applications of computer vision vary, but a typical vision system uses a similar sequence of distinct steps to process and analyze image data.

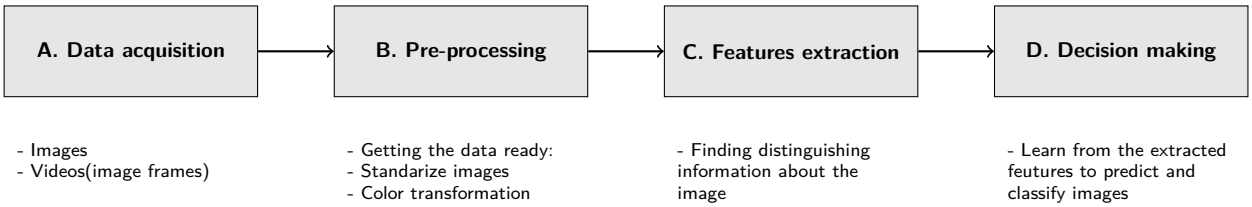


Figure 1.4: Typical steps in computer vision systems [3].

Figure 1.4 shows that computer vision applications start off by acquiring images data, then processing these data, performing some analysis and recognition steps, and finally make a prediction based on the extracted information [40, 41].

1.3.2.2.1 A. Data acquisition

A digital image is produced by one or several image sensors which are cameras, scanners, camera drones, and satellites, and more. The image taken is an array of pixels .

1.3.2.2.2 B. Preprocessing

Preprocessing is a common name for image operations at the lowest level of abstraction, both input and output are intensity images. The aim of preprocessing is to improve and enhance the image data that suppresses unwilling distortions or enhances the quality of some image features although geometric transformations.

1.3.2.2.3 C. Feature extraction

Features help to define certain objects, they are usually information about object shape or color. Image features extracted can be lines, ridges, edges, points, blobs, corners, or motion texture [41].

1.3.2.2.4 D. Decision making

Making the final decision required for the application. Through the features extracted from the previous step, predicting the class of image.

1.3.3 Image Recognition

IR is a higher-level process that requires more intelligence than other parts of image processing such as image enhancement. [42]. And it is the ability of software to identify objects, places, people, writing, and actions into images [43]. Different varieties of the recognition problem are object recognition, detection, and identification [44].

- **Object recognition**: called also object classification one or several object classes can be recognized, usually together with their 2D positions in the image or 3D poses in the scene.
- **Detection**: is finding smaller regions of interesting image data based on relatively simple and fast computations.
- **Identification**: recognizes an individual instance of an object.

1.4 Technical methods

Artificial intelligence is a wide-ranging branch of computer science. It involves any technique that aims to enable computers to mimic human behavior, including machine learning and deep learning [45]. In the next part, we will mention machine learning, deep learning, in addition we defining the standard metrics to evaluate classification, and explaining the most important differences between them.

1.4.1 Machine learning

Machine learning is a subset of AI techniques that enables computer systems to learn from previous experience, i.e. data observations, and improve their behavior for a given task. It focuses on the development of computer programs that can access data and use it to learn for themselves [45, 46, 47]. Machine learning algorithms are often categorized into four types:

- Supervised machine learning.
- Unsupervised machine learning.
- Semi-supervised machine learning.
- Reinforcement machine learning.

1.4.1.1 Supervised machine learning algorithms

Supervised learning considers learning to be guided by a teacher. There is a dataset that acts as a teacher and its role is to train the model or the machine [48], the dataset is a collection of related sets of data arranged in a format, that is composed of separate elements but can be manipulated as a unit by a computer [49, 50]. Figure 1.9 represents the mechanism of supervised learning.

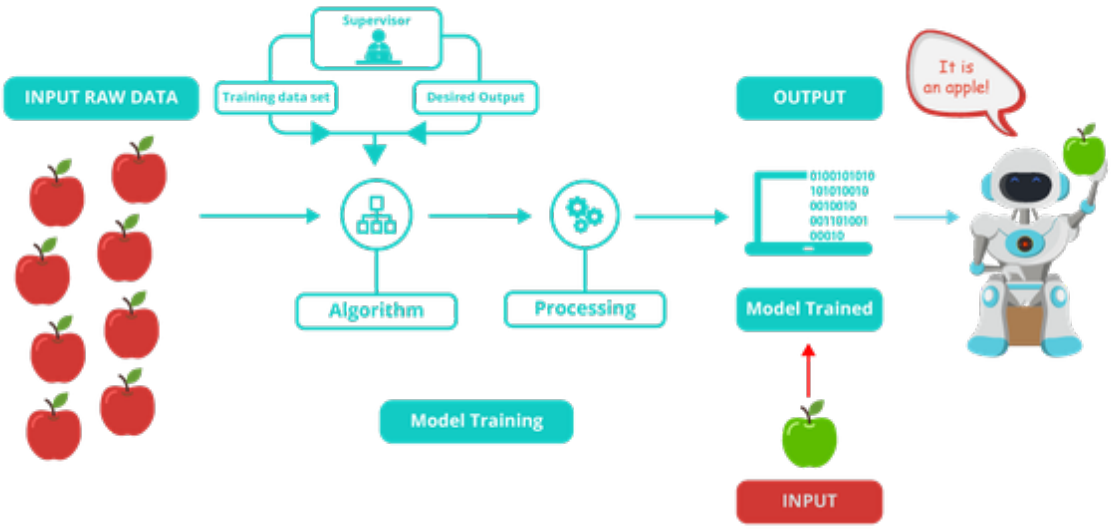


Figure 1.5: Supervised learning [4].

The important algorithms are:

A. Categorical: regression (Logistics, Neural Networks,..), classification (KNN, Trees, Naive Bayes, SVM,..).

B. Continuous: regression (Leaner, Polynomial..), classification (Decision tree, Random forest..).

We take Neural Network, Support vector machines, and K-Nearest Neighbors as examples of supervised learning algorithm.

1.4.1.1.1 Neural Network

NN is artificial neural networks that are a subset of machine learning techniques, loosely inspired by biological neural networks. They are usually described as a collection of connected units, called artificial neurons, organized in layers [51]. As shown in figure 1.6.

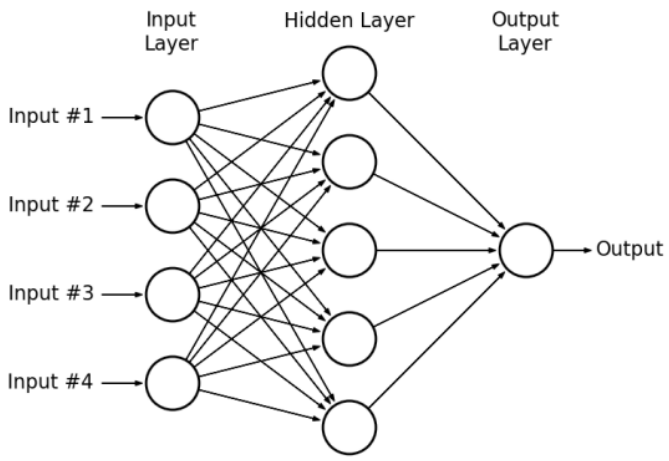


Figure 1.6: Simple multi-layer neural network structure [5].

1.4.1.1.2 Support vector machines

SVMs are a set of related supervised learning methods that are applicable to both classification and regression problems using data analysis and pattern recognition. A support vector machine creates a maximum-margin hyperplane or set of hyperplanes to classify all inputs in a high-dimensional space or even infinite space [52, 53]. The architecture of the SVM is shown in figure 1.7.

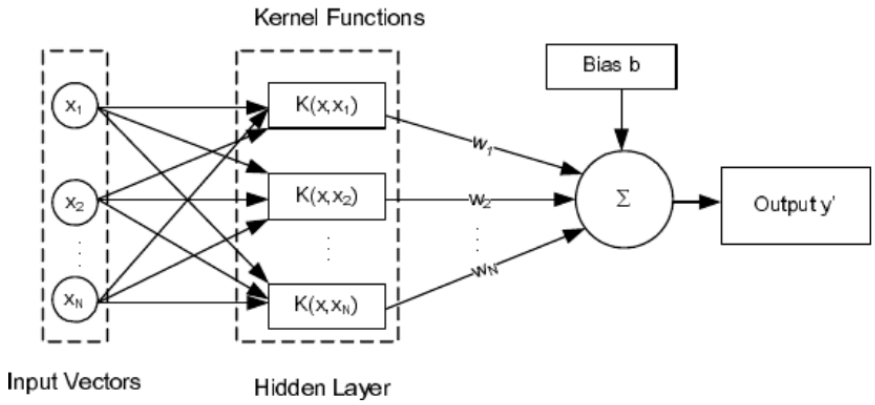


Figure 1.7: Architecture of a Support Vector Machine classifier [6].

1.4.1.1.3 K-Nearest Neighbors

K-NN classifier is one of the prospective statistical classification algorithms, it is a conventional non-parametric classifier used for classifying objects based on the closest training data set in the feature space. The K-Nearest Neighbors of an instance is defined in accordance with Euclidean distance [54, 55, 56].

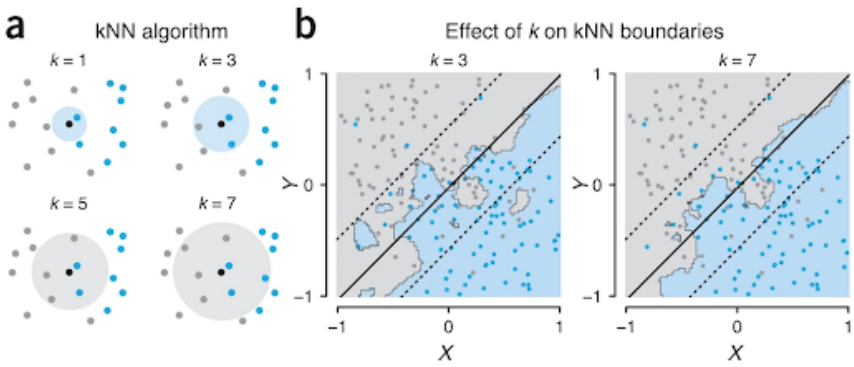


Figure 1.8: Illustration of the k-nearest neighbors (kNN) classifier [7].

1.4.1.2 Unsupervised machine learning algorithms

The model learns through observation and finds structures in the data. Once the model is given a dataset, it automatically finds patterns and relationships in the dataset by creating clusters in it[57], as shown in figure 1.9. The important algorithms are:

a. **Categorical**: association (Apriori, FP-Growth,...), Hidden Markov Model,. . . etc.

b. **Continuous**: clustering (SVD, PCA, K-means,. . .).

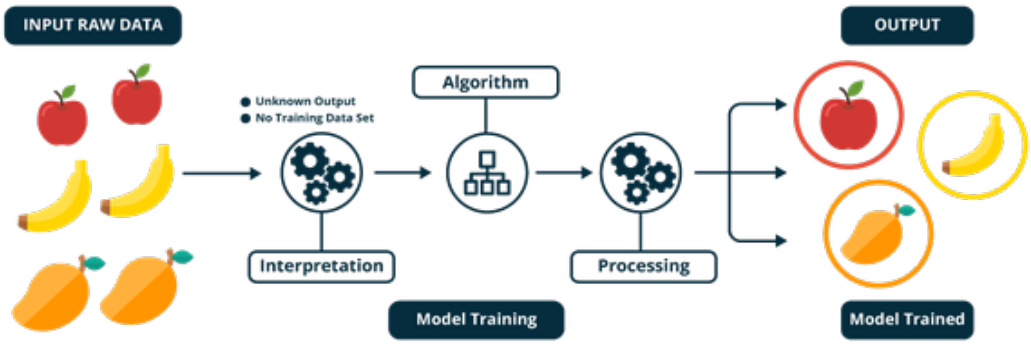


Figure 1.9: Unsupervised learning [4].

1.4.1.3 Semi-supervised machine learning algorithms

Falls between unsupervised learning (with no labeled training data) and supervised learning (with only labeled training data), that combines a small amount of labeled data with a large amount of unlabeled data during training [58, 59]. Figure 1.10 shows the mechanism of the semi-supervised learning technique.

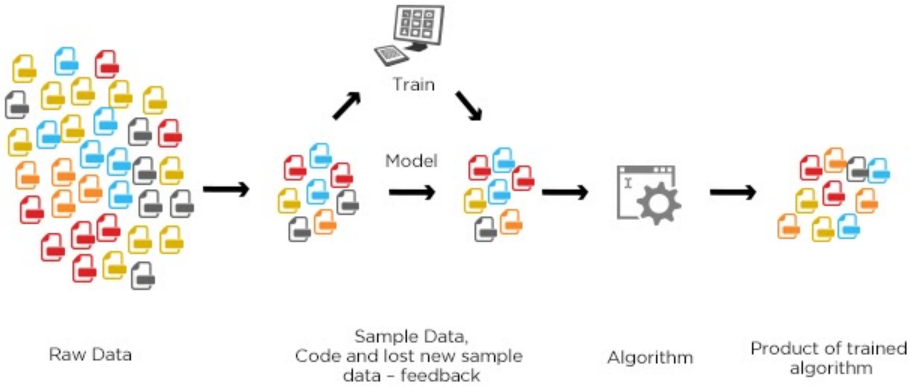


Figure 1.10: Semi-supervised learning [8].

1.4.1.4 Reinforcement machine learning algorithms

The ability of an agent to interact with the environment and find out what is the best outcome. It follows the concept of the hit and trial method [47]. Figure 1.11 represents the mechanism of reinforcement learning by interacting with the environment.

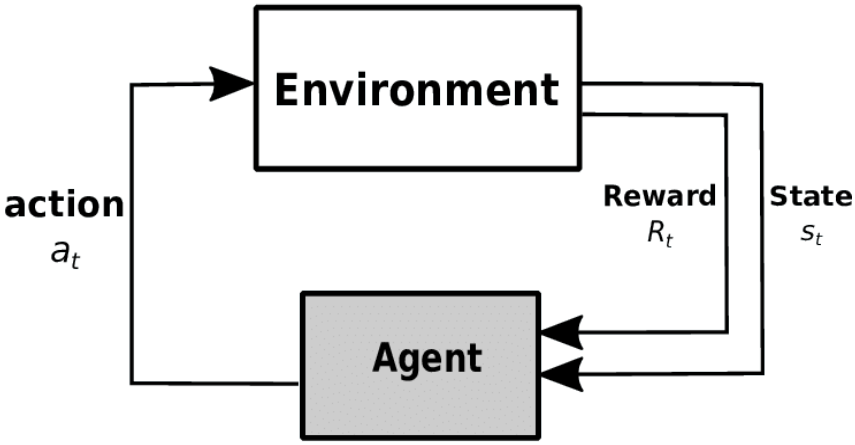


Figure 1.11: Reinforcement Learning, Agent and Environment [9].

1.4.2 Deep Learning

DL is a subset of artificial intelligence that makes the computational multi-layer neural network feasible as shown in figure 1.12.

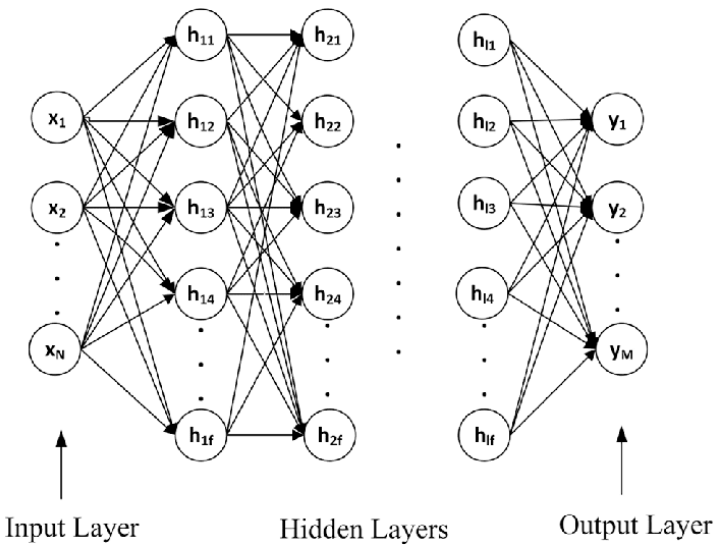


Figure 1.12: Typical deep neural network [10].

Typical deep learning architectures are deep neural networks, convolutional neural networks, recurrent neural networks, generative adversarial networks, and more [51]. In this part, we mention convolutional neural networks, then we review the pretrained models.

1.4.2.1 Convolutional Neural Networks

Convolution is an integral transformation operation on a function through a specific operator. It reforms the original function into a new representation of the similarity of two functions in a certain window, and it is one of the most used operations in deep learning for computer vision [60, 61].

CNN is a type of deep learning network that has become popular for image classification. It is simply a neural network that uses convolution in place of general matrix multiplication in at least one of their layers [62, 63].

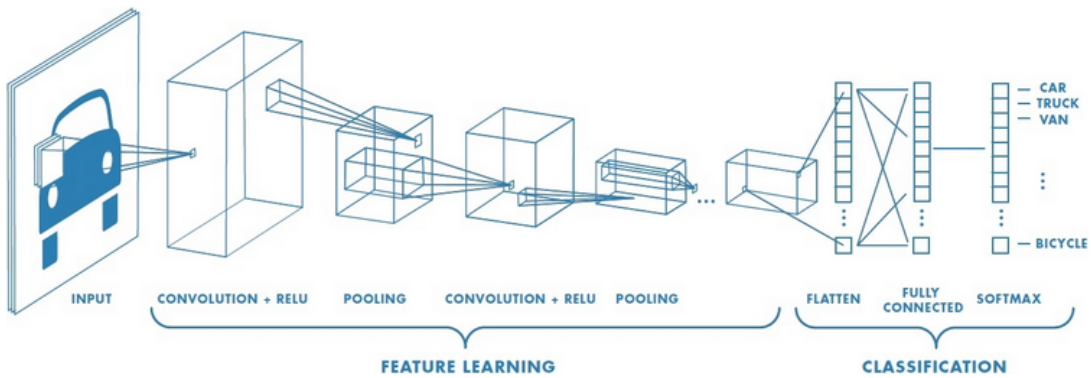


Figure 1.13: A network with many convolutional layers [5].

Figure 1.13 shows CNN layers perform operations that alter the data to learning features specific to the data. Three of the most common operations are convolution, and

activation (ReLU), and pooling [62, 64, 65].

- **Convolution operation** puts the input images through a set of convolutional filters, each of which activates certain features from the images.
- **Activation operation**(Rectified linear unit function) allows for faster and more effective training by mapping negative values to zero and maintaining positive values. Sometimes, it is referred to as activation because only the activated features are carried forward into the next layer.
- **Pooling operation** simplifies the output by performing nonlinear downsampling, reducing the number of parameters that the network needs to learn [62].

These operations are repeated over tens or hundreds of layers, with each layer learning to identify different features [65]. The classification layer is the one before the output layer. It is a fully connected layer with N-dimensional output, N being the number of classes to be categorized. This layer outputs an N-dimensional vector, each element of the vector is the probability that the input image belongs to one of the N classes. The final output layer uses a softmax function to give the classified output. To produce accurate results, thousands or millions of images have to be fed into the neural network [62].

1.4.2.2 Pretrained models

A pretrained model is a saved network that was previously trained on a large dataset. There are two uses of pretrained models that use the pretrained model as-is for prediction or uses transfer learning to customize this model to a given task. Accordingly, due to the computational cost of training any models, it is used models that indeed were trained [66, 67].

1.4.2.2.1 Transfer Learning

It is a deep learning technique that enables developers to harness a neural network used for one task and apply it to another domain. There are two ways to customize a pretrained model [66, 68]:

1. **Feature Extraction**: use the representations learned by a previous network to extract meaningful features from new samples.
2. **Fine-tuning CNN models**: relies on retaining the layers of the pre-trained CNN model which is responsible for feature extraction. In transfer learning, this first step is to place a set of new layers able to classify new classes.

1.4.2.2.2 Pretrained models examples

Pretrained models are models built by various organizations or individuals as part of their research work or as a part of competition. These models are generally very complex and are trained on large amounts of data. Such pretrained models can be reused through the transfer learning paradigme.There are several pretrained models made available by various parties [69]. The following described are some of pretrained models.

- **VGG 16 and VGG19**

  The VGG is a was one of the famous models submitted to ILSVRC-2014 (an annual competition called the ImageNet Large-Scale Visual Recognition Challenge) [70, 71, 72]. This network is characterized by its simplicity, using only 3×3 convolutional layers stacked on top of each other in increasing depth. Reducing volume size is handled by max pooling. Two fully-connected layers, each with 4,096 nodes are then followed by a softmax classifier. The number "16" and "19" stand for the number of weight layers in the network [70, 72].

- **ResNet 50**

  First introduced in the Deep Residual Learning for Image Recognition paper. ResNet is instead a form of exotic architecture that relies on micro-architecture modules, i.e. network-in-network architectures. It uses blocks that reroute the input and add to the concept learned from the previous layer. During learning the next layer will learn the concepts of the previous layer plus the input of that previous layer. This would work better than just learn a concept without a reference that was used to learn that concept [73, 74, 75].

- **Inception V3**

  The Inception micro-architecture was first introduced in Going Deeper with Convolutions paper. The goal of the inception module is to act as a multi-level feature extractor by computing 1×1, 3×3, and 5×5 convolutions within the same module of the network. The output of these filters is then stacked along the channel dimension, before being fed into the next layer in the network [72, 74].

- **Xception**

  Xception is an extension of the Inception architecture which replaces the standard Inception modules with depthwise separable convolutions. It introduced in the CVPR paper, Xception: Deep Learning with Depthwise Separable Convolutions. Beside it is an efficient architecture that relies on two main points, Depthwise Separable Convolution, and shortcuts between Convolution blocks as in ResNet. The network has learned rich feature representations for a wide range of images [76, 72].

The following deep convolutional neural networks: VGG (VGG16 and VGG19), GoogLeNet (Inception V3 and Xception), and ResNet (ResNet50) architectures, shown in figure 1.14, pre-trained for object detection task on the ImageNet dataset [11].

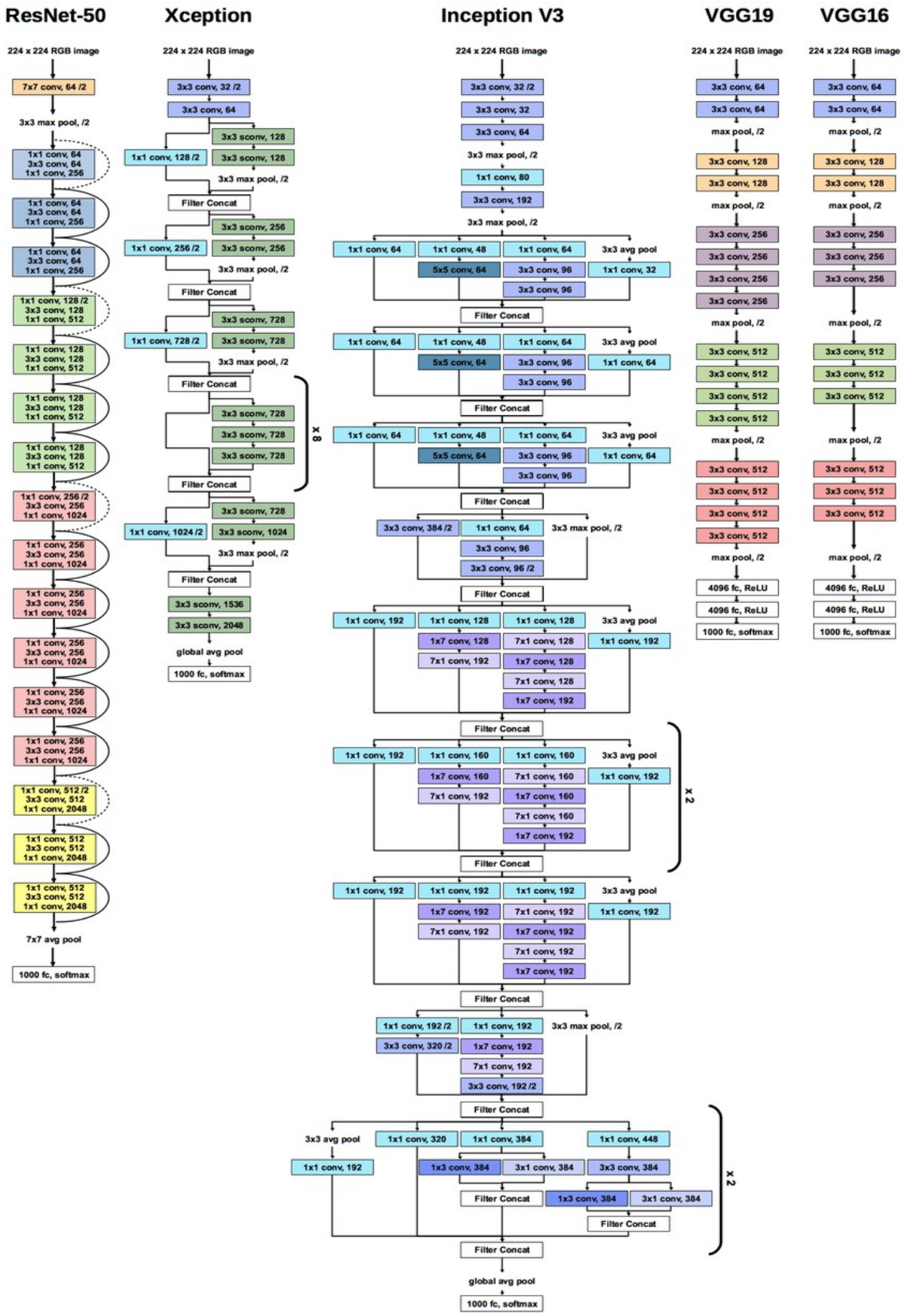


Figure 1.14: VGG16, VGG19, Inception V3, Xception and ResNet-50 architectures [11].

1.4.2.3 Model Evaluation Metrics

The most widely-used evaluation metric for image recognition models is the accuracy of test data because the accuracy can be calculated efficiently, they often correlate well with many standard metrics such as a confusion matrix, precision, F1-score, and recall [77, 78]. In this part, we attempt to see these metrics.

1.4.2.3.1 Accuracy

Accuracy is the fraction of predictions the model got right. Formally, accuracy is the percentage of observations correctly classified. For binary classification, accuracy can also be calculated by positives terms of and negatives term, as follows:

$$Accuracy = \frac{Number_of_correct_predictions}{Total_of_predictions}$$

Where:

- TP = True Positives.
- TN = True Negatives.
- FP = False Positives.
- FN = False Negatives.

1.4.2.3.2 Confusion matrix

A confusion matrix is a specific table layout. Each row of the matrix represents the instances in a predicted class, while each column represents the instances in an actual class (or vice versa), this table that allows visualization of the performance of an algorithm [79].

1.4.2.3.3 Precision

Precision is the percentage of correctly classified as positive in the group that the classifier has declared positive. It is calculated by the next equation:

$$Precision = \frac{TP}{TP + FP}$$

1.4.2.3.4 Recall

Recall this is the percentage of observations labeled as positive that are correctly classified. It is calculated by the next equation:

$$Recall = \frac{TP}{TP + FN}$$

1.4.2.3.5 F_1-score

F_1-score also called F 1-measure, is the measure that strikes a balance between precision and recall. It is computed as the harmonic mean of the precision, and recall with the value ranging between 0 (worst score) and 1 (best score). It is calculated by the next equation:

$$F_1\text{-score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

1.5 Comparison of deep learning and machine learning

In this part, we want to highlight the important differences between machine learning and deep learning. Table 1.1 outlines the main points from them.

Approach used	**Machine learning**	**deep learning**
Training dataset	Small	Larg
Choose features	manually	automatically
Classifiers	need to be tried for best results	applied on the model automatically
Training time	Short	Long
Accuracy	Limited	Illimited

Table 1.1: Comparison of deep learning and machine learning [15].

Deep learning is a subtype of machine learning. In machine learning, we manually choose features and a classifier to sort images. While deep learning extract and model the features automatically. Deep learning needs lots of data to produce the best results and is computationally intensive. Machine learning and deep learning offer a variety of techniques and algorithms that can be choose based on many factors, the size of data we're processing, and the type of problem want to solve [15].

Conclusion

In this chapter, we present basic concepts about the human foot trace recognition, the technical disciplines in the visual system, and from the artificial intelligence branch, we mention a technique that aims to enable computers to mimic human behavior, including machine learning and deep learning. In the next chapter, we will discuss human identification methods from footmarks in forensic podiatrists filed.

Chapter 2

Human identification from foot marks

Introduction

The foot is an anatomical region that it can identify individually human from nonhumans or even human from each other. Also, it is possible to identify a person from considering the marks left by the foot on its surrounding environment. In this, we must consider different scenarios where the foot prints or marks may be on the ground or other surface, and with or without shoe. Unknown foot marks at such sites would be compared directly with the footprints of a known individual and a conclusion reached as to whether these marks shared common ownership. Fundamentally, identification process uses plantar prints or impressions. Several techniques have been derived, but initially use various anatomical measurements (heel-to-toe measurements, the angles between these measurements, and across-foot distances) to describe foot marks. These measurements can then be compared with other marks (or footwear) to obtain a potential owner.

Currently, there are four main sub-specialties within the scope of practice of forensic podiatrists:

1. Analysis and Identification involving Podiatry Treatment Records
2. Barefoot prints
3. Footwear
4. Gait Patterns

The three last elements are the only once with relation with barefoot identification.

2.1 Barefoot prints comparison methods

An individual's footprints are unique as fingerprints and can be used for human identification purposes. Even monozygotic twins have different footprints; which means once being able to find similarities it is a great step towards identifying possible people. This kind of identification process is much more used where it is common to walk barefoot [12]. To collect a proper bare footprint, the suspect must walk

uninterrupted along 6 meters. Also, two podiatrists must control the process to make sure there is a good quality of footprints. Perfect footprint must include: 5 toes, ball of foot impression, heel impression, lateral mid-foot impression and arch profile. With all of these features, the size, the orientation and the shape help the identification [13, 80, 14].

Several techniques have been employed for barefoot identification, which include:

1. The Gunn method;
2. The Optical Center Approach;
3. The Overlay method;
4. The Robbins method;
5. The Rossi's podometric system.

2.1.1 The Gunn Method

This method is developed by the Canadian forensic podiatrist Dr. Norman Gunn in 1972. It is used to assess, compare, and evaluate plantar prints. The basic idea is to draw five lines from the rearmost aspect of the heel print to the tip of each toe print. After that, another cross line is drawn across the widest aspect of the ball of the foot area (see figure 2.1) [81, 82].

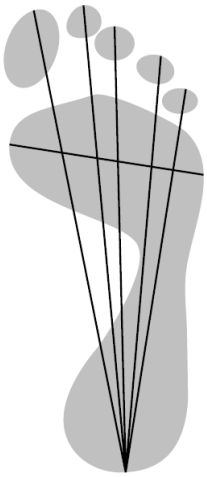


Figure 2.1: The Gunn Method [12].

These lines measures are used to compare two different footprints or marks. In addition to enforce comparisons, angles between lines are used. The basic measurements and angles used in the Gunn Method are [13]:

1. Heel to first toe;
2. Heel to second toe;
3. Heel to third toe;
4. Heel to fourth toe;

5. Heel to fifth toe;
6. Heel to first toe line — intersection with cross ball line;
7. Heel to second toe line — intersection with cross ball line;
8. Heel to third toe line — intersection with cross ball line;
9. Heel to fourth toe line — intersection with cross ball line;
10. Heel to fifth toe line — intersection with cross ball line;
11. Angle between heel to first toe and heel to second toe lines;
12. Angle between heel to first toe and heel to third toe lines;
13. Angle between heel to first toe and heel to fourth toe lines;
14. Angle between heel to first toe and heel to fifth toe lines.

Figure 2.2: Development of the Gunn method[13].

2.1.2 The Optical Center Approach

This approach represents an evolution from the precedent one to evaluate more details in footprint. It was developed by the Royal Canadian Mounted Police in the aim of demonstrating the individuality of human footprint. They used a computer database composed of 24000 human footprints to compare and evaluate different footprints using the optical center approach. The latter, consists of determining the central point of a morphological feature as defined by the center of a circle when placed in a position of best fit (see figure 2.3)[13, 83, 12].

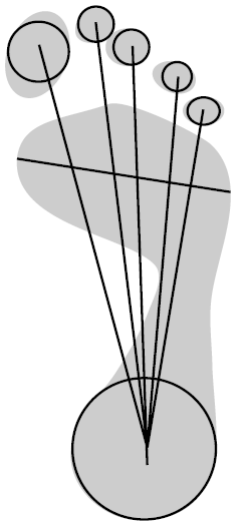


Figure 2.3: The Optical Center Method [13].

This position is defined using a computer software. It starts by defining a set of concentric circles with a set of marked central positions. Then, a set of basic lines are drawn from the defined center of the heel print to the optical centers of each toe print, also, like the precedent method another line is added across the widest area of ball of the foot area. Finally, a set of measurements from different lines (from point to point) and angles are taken to compare different footprints. This approach permits to obtain a total of 38 variable measurements [14, 12].

2.1.3 The Overlay Method

This method is completely a different approach compared to other methods developed in Canada. It was developed and used by the Forensic Science Service in the U.K. In this case, an outline of a known footprint is placed onto a clear acetate sheet, and this image is laid over a questioned print [80, 82].

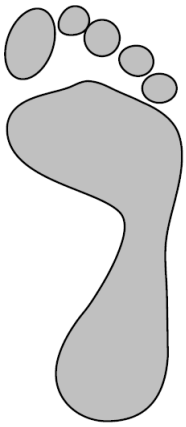


Figure 2.4: The Overlay Method [14].

This approach uses the morphological outlines of different footprints instead of point to point measurements to compare between two foot marks. During the process, the outline

shape (contour) of the footprint being analyzed is compared with the outline shape of another footprint (a known footprint or mark)[13, 14].

2.1.4 The Robbins Method

This method is very similar to Gunn and Optical Centers methods, but, with some additional measurement lines. It was developed by The forensic anthropologist Louise Robbins in 1985. This technique uses a set of lines to extract visual anthropological measurements, some of them are similar to those drawn by the Gunn method. In addition, some right angled lines are added to extract additional measurements. Those lines are drawn from a theoretical base line across the rear of the heel to the tips of each toe. Also, a declination angle is drawn between the tips of the first toe to fifth toe (see figure 2.5) [13, 12].

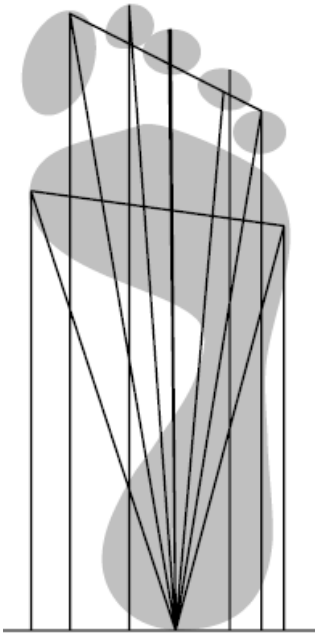


Figure 2.5: The Robbins Method[13].

2.1.5 The Rossi's Podometric System

In 1992, the American podiatrist William Rossi developed an alternative approach of categorizing landmarks of the foot. The method was based on drawing triangular lines longitudinally and transversely across the foot print. These lines create patterns that permits to categorize a given footprint [13, 82]. The first line is drawn to connect the outermost lateral aspect of the heel print with the lateral aspect of the ball of the foot area. While the second line connects innermost medial aspect of heel with the medial aspect of the ball of foot area. After that, three other centered lines are added in between the existing lines. Similarly, other five lines are added laterally (see figure 2.6). The comparison of two footprints or marks uses the angles and distances to record and

quantify the different dimensions [14, 12].

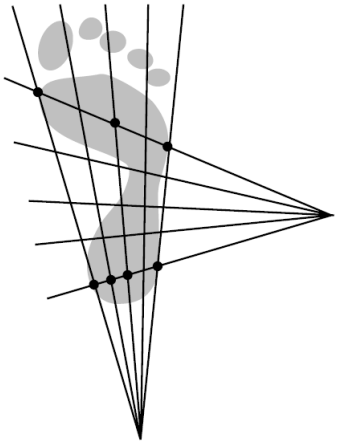


Figure 2.6: Rossi's system of podometrics[12].

The five mentioned methods have been used in different circumstances to identify human footprints. However, each method has its strengths and weaknesses[13]. The table 2.1 illustrates the different aspects of each one them.

Standards \ Methods	Gunn	Optical Center	Overlay	Robbins	Rossi
Allows quantitative measurement	Yes	Yes	No	Yes	Yes
Allows detail comparison	No	No	Yes	No	No
Repeatability validated	No	Yes	No	No	No
Ease of use	Yes	No	No	Yes	Yes
Considers footprint as a whole	Yes	Yes	No	Yes	Yes
Frequency of use in footprint identification	Yes	Yes	Yes	No	No
Use in dealing with partial footprints	No	No	Yes	No	No

Table 2.1: Different aspects of barefoot prints comparaison methods.

2.2 Identification involving footwear

This kind of identification has been used for a long time to find criminals via footwear prints. The process includes finding the shoeprints at a crime scene, which could be linked to the outsole of the shoe that made these prints. The latter, is matched with another shoe print of a possible suspect [13, 14, 84].

The foot impression within the shoe with the different techniques mentioned above (Gunn, Optical Center, Overlay method, ...etc.) are the key to link an individual to a given pair of shoes. In addition, other factors must be included to be able to deal with the different variables included by wearing a pair of shoes [14, 84].

There are three possible ways to obtain information from shoes.

1. Outer impressions;

2. Inner impressions;

3. Other evidence from shoes.

2.2.1 Outer impressions

The examiner should observe the footwear from all different angles in order to extract a set of characteristics, which include: detail of the style, indicated make, shoe type, color, and marked size. Also, other descriptors like distortions, crease lines of the upper as well as the presence of any unusual features such as areas of deliberate damage and style of lacing should be recorded [14, 84].

Using focal point, it is possible to determine a set of areas that can describe the outsole wear for later comparison purposes. Areas 1to 9 are located in the heel and areas 10 to 21 are located in midfoot and forefoot (see figure2.7)[14, 84].

Figure 2.7: Focal point instrument[14].

2.2.2 Inner impressions

It presents the imprints that have been left in the inside of a shoe by contact from the person's foot. These impressions can be used to link a person to a piece of footwear. Also, finding body hair, body fluids, skin tags, dust particles, glass fragments inside a shoe presents a very helpful information or evidence to link a footwear to a location or owner [13, 14].

Figure 2.8: Focal point instrument [14].

2.3 Forensic Gait Analysis

Gait or gait trait can be a very helpful to recognize or identify a person. Forensic gait analysis, is a very valuable technique in identification. The basic idea is that it is possible to identify a person by their unique characteristic manner of walk. The gait can be divided into two sections; The Stance Phase (foot is in contact with the ground) and The Swing Phase (foot is not in contact with the ground) [13, 14]. Also, it can be viewed as the interpretation and quantification of human movements including: foot and ankle, knee and hip. Two possible ways of evaluation are used [14, 84]:

1. **Qualitative:** The use of non-numerical data used for the evaluation of movement. A visual assessment is needed with sagittal, frontal and rear views.
2. **Quantitative:** In this case technology is needed to analyze the collected numerical information, which include forces, pressure and movements.

The examination process uses a set of events recordings, which have been captured on closed circuit television footage, to identify any features of interest shown by the person. Then, it is possible to compare these features with other features from other known recordings [14].

Conclusion

For many years the profession of forensic podiatry has been existing but it has gain recognition only from a short time. Forensic podiatrists are professionals specialized in feet that have expertise level related to the foot. There are four subspecialty roles for a forensic podiatrist which can help to cover and analyze podiatric records to identify a barefoot print, or understand a footwear print and finally, give a possible recognized person or suspect.

Chapter 3

Design and implementation

Introduction

In the previous chapters, we presented an overview on foot trace recognition, technical disciplines, and methods used in the process of human feet identification. In this chapter, we conduct a comparative study of two different systems architectures. Both systems use deep learning methods, for classification (recognition) of the HFT images. For this purpose, we present the general steps of the design process for the HFT system. These steps are common to both proposed systems. Then, we show the detailed design steps for each one.

In this section, we explain the important general steps in the HFTs design process. The HFTs design process involves eight steps, images data acquisition, preprocessing, dataset construction, dataset preparing, features extraction, classification, matching, and making decisions. As shown in figure 3.1. These steps are common to each system.

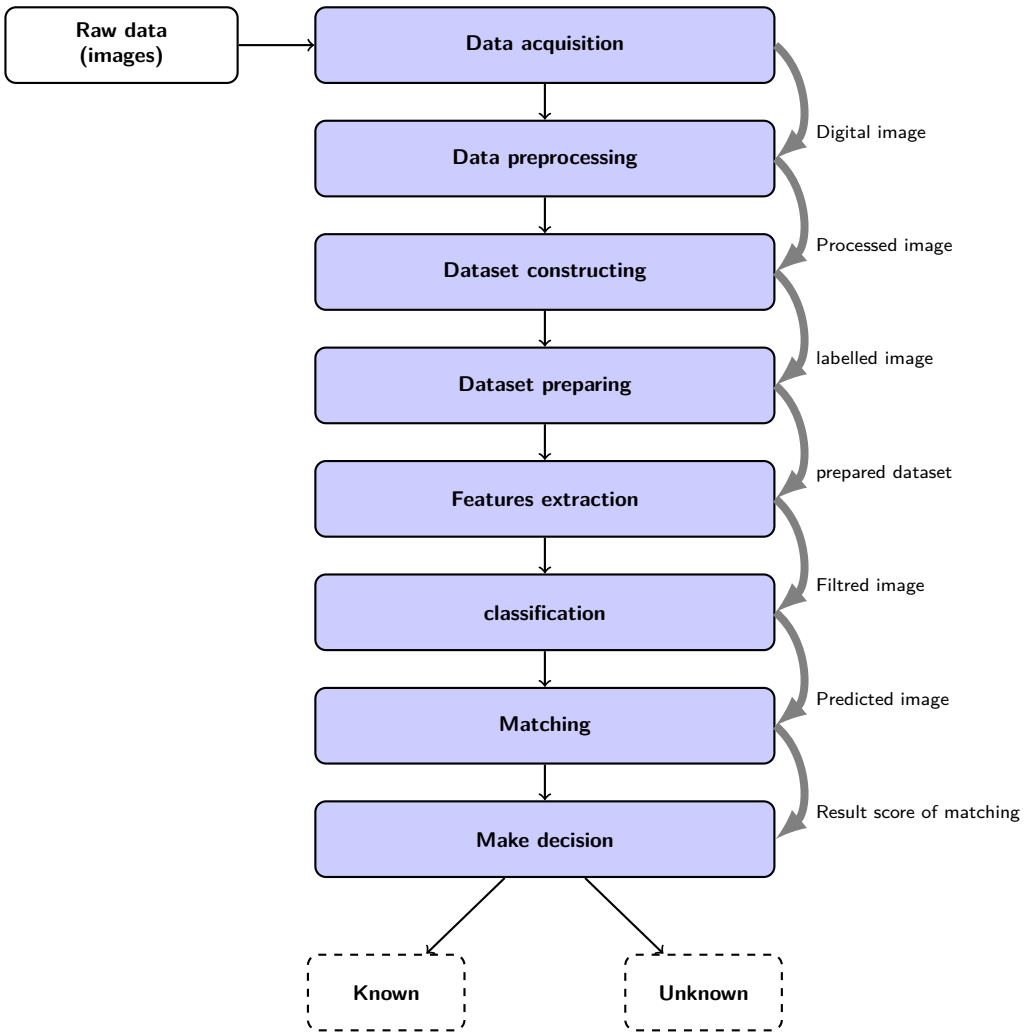


Figure 3.1: General HFTs design process.

We categorize these eight steps into two main stages, the dataset generation and the systems model (s) construction. In the first phase, we generate and prepare data sets from primitive data to train our proposed models. For the second phase, we build the models through training and testing steps.

3.1 Dataset generation

In this phase, we collect and process a set of raw data to be clear and clean. Based on the processed data, we generate a dataset for each proposed model. We conclude this stage by preparing the generated datasets for the models.

3.1.1 Data acquisition

In this step, we need to obtain an HFT dataset from the famous websites of the dataset. But, the unavailability of the dataset is one the biggest barrier to the success of deep learning systems. As in our case, the HFT dataset is unavailable on the most famous

websites such as Kaggle, Dataset Search, Socrata, and Google Public Datasets... etc. Therefore, on 30 January 2020 along 6 days, we collected a set of HFT images personally from 102 individuals (volunteers), at El-Oued, Algeria. Each person has the most necessary information, e.g. ID, full name, tall, weight, age, as shown in figure 3.2.

id	name	tall	wheight	age	type_foot	barefoot	footweare	age_group
1	Emirat meriem	1.65	50	22	O	T	T	A
2	rezzag lobza najah	1.67	65	28	O	F	T	A
3	aggab sahar	1.62	65	24	O	T	T	A
4	mohammed orwa	1.50	40	14	X	T	F	T
5	larousi tahraoui	1.48	44	13	II	F	T	T
6	raeid ben abdellah	1.58	52	14	II	T	T	T
7	Ghamam abdelhay	1.2	33	9	II	T	F	K
8	abdelkader gitoubi	1.82	61	19	II	F	T	A
9	nouajea mouhammed arbi	1.70	68	19	II	F	T	A
10	abderrahman elgayed	1.78	59	18	II	F	T	T
11	roumaysa shebata	1.65	62	18	II	F	T	T
12	narjes guadaa	1.65	60	18	X	F	T	T
13	aade elaid	1.83	74	19	X	T	F	A
14	gherissi ammar	1.65	62	21	X	T	F	A
15	othmani ayoub	1.74	65	19	II	F	T	A
16	chabrou mohammed	1.70	60	18	II	F	T	T

Activer Windows
Accédez aux paramètres

Figure 3.2: Volunteers information.

The total size of the collected data is 7 GB. Figure 3.3 shows the collected raw data without processing i.e. the raw data is not cleaned, or filtered, or enhancing ...etc.

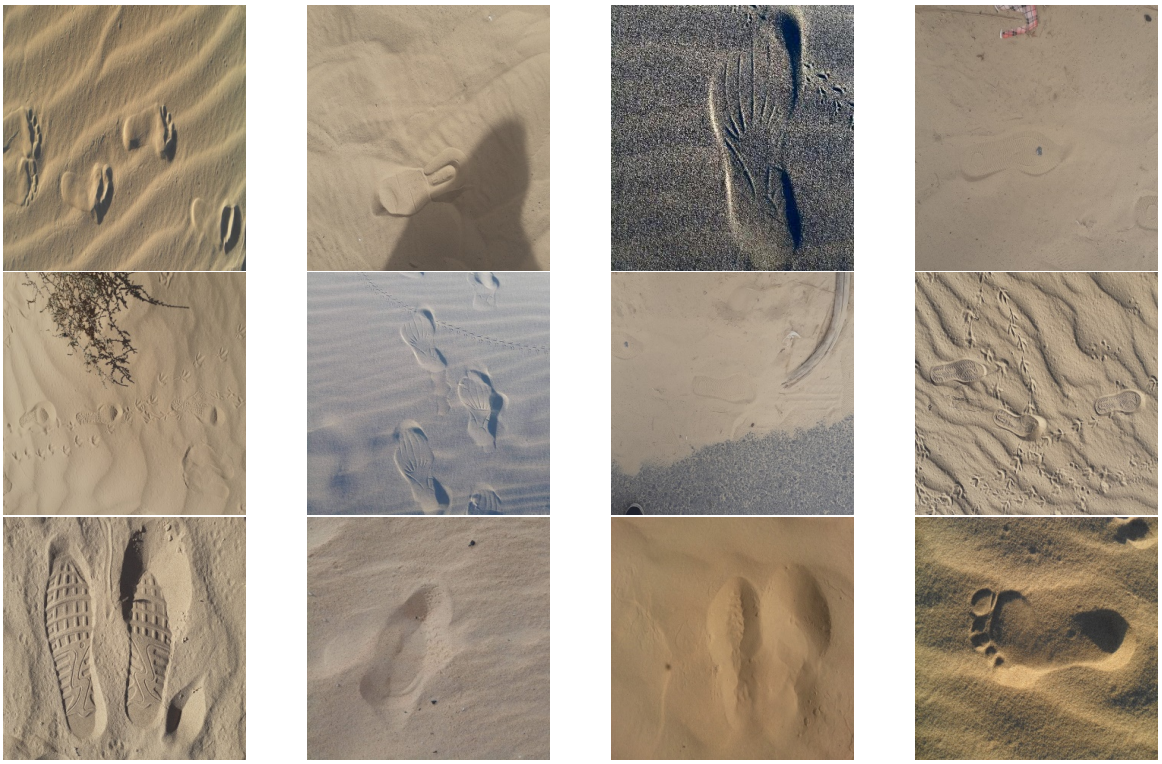


Figure 3.3: Sample of collected raw data.

For each one, we focus on taking images with some situations which are footwear or/and barefoot, two feet together, right foot, left foot, and a set of footsteps. These images are taken by three devices, two Samsung phones, and SAMSUNG digital imaging device:

- SAMSUNG phone 1 (model: M20, Resolution image: 4128x3096).
- SAMSUNG phone 2 (model: GT-N7100, image resolution: 2448x 3264).
- SAMSUNG digital camera device (model: PL 101, image resolution: 4000x3000).

When we were collecting data, some factors were taken into consideration. These factors are taken to preserve image quality and characteristics. the factors are:

- Natural lighting (the sun).
- Sand quality.
- No shadow reflection of anything in the field of work.
- The sand should not be so high, to not erase the traces, so the traces appear as immersed in the sand which leads to the loss of the most important features of the foot and the shape of its edges.

3.1.2 Data preprocessing

Image pre-processing is an important step as it helps to enhance the quality of the image for further processing. The preprocessing step is necessary as an image inherently contains noise and this may result in lower classification accuracy. This step is performed to remove the noise that hampers the identification process and handle the degraded data. It goes through three steps of preprocessing, selecting, cleaning, and enhancement.

3.1.2.1 Data selection

Data selection is selecting the cleared image and excluded (removed) uncleared image from the data. After the data was collected, we selected from it a set of images that showed the foot traces more clearly, otherwise, we excluded the rest of the data images that did not select because it was difficult to distinguish the foot traces from the sand. In this step, all data images were selected carefully and separately. Figure 3.4 shows a sample of the selected images (a) and excluded images (b).

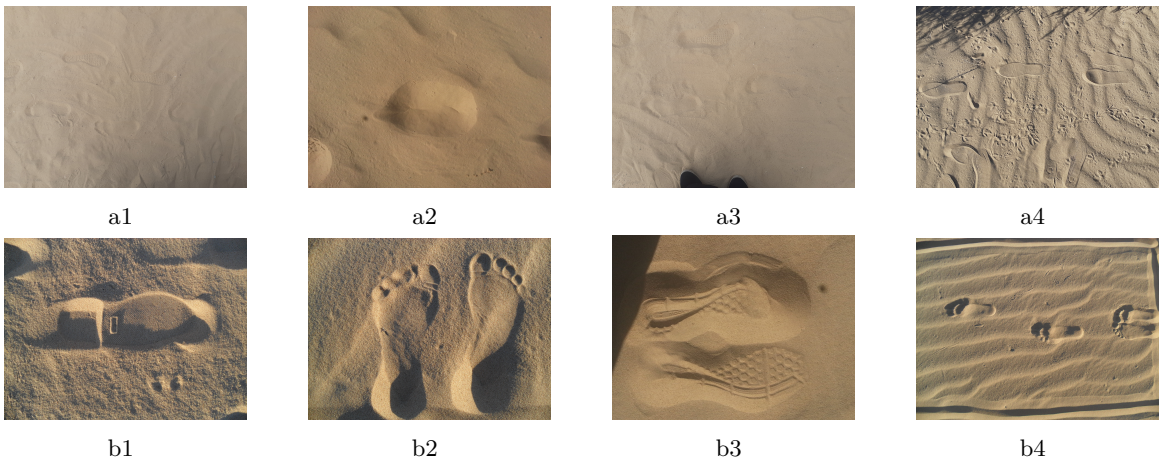


Figure 3.4: Sample of the selected (a) and excluded (b) images.

3.1.2.2 Data cleaning

The process of cleaning the images was done manually to ensure that the images had not misleading traces or impurities. So, we cut some parts of images because it includes many other traces and objects outside the foot traces of volunteers, such as traces of small insects, plant leaves, or gravel... etc. We filtered any impurities that might affect the quality and features of the image. Figure 3.5 is shown samples of the images including misleading traces of impurities.

Figure 3.5: Foot traces not cleaned.

After cutting all unnecessary parts in order to facilitate the automatic treatment later, we get clean data. Among prominent defects and impurities found:

- Sand contains impurities such as Halfa plants [85], gravel, and waste.
- Sand contains other traces, such as traces of other persons, animals, or birds, Insects, reptiles ... and so forth.
- Shadow reflection in the field of work.

- Traces immersed in the sand.

Figure 3.6 is shown samples of the images foot traces selected and cleaned.

Figure 3.6: Sample of cleaned data.

3.1.2.3 Data enhancing

After we select and clean the data images, we try several ways to improve the quality of each image. Therefore, we analyzed the images by using a histogram of the pixel intensity values, to study the possibility of applying filters of enhancements on these images. In this step, we analyze the histogram of the pixel intensity values for the image in three different cases (RGB color, GrayScale, Binary). Depending on image condition and type, we nominated the best case as a result of comparing the histograms image in three cases. In order to later apply the filter of enhancement to the image in the best state, which is increasing the percentage of image quality improvement.

3.1.2.3.1 RGB color enhancing

In our state, the images collected are in BGR format. Therefore, we convert it to an RGB format to get the RGB color histogram. We convert the image to an RGB format by using the cvtColor function. In this part, we will show the histogram of color images (pixels intensity). The pixel intensity value is represented by three axes RGB, i.e. red, green, and blue channels [86]. Figures 3.7, 3.8,3.9 are shown the histograms of three axes RGB for color images.

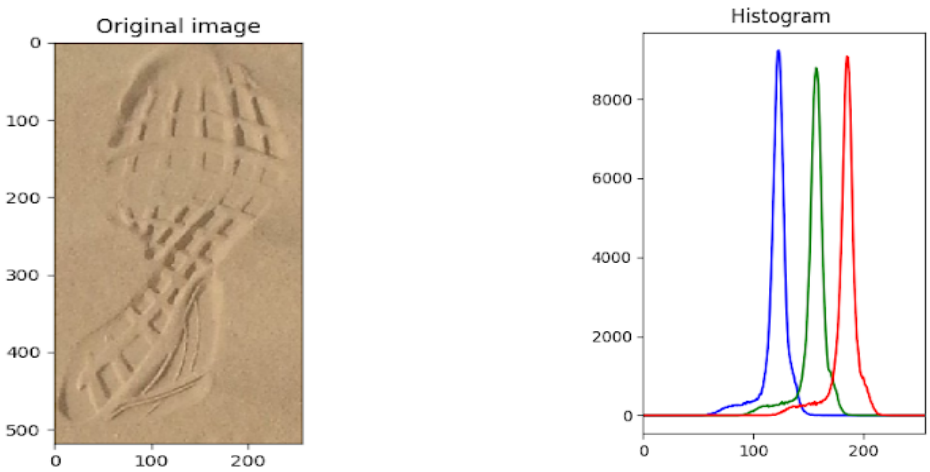


Figure 3.7: Image RGB color of footwear trace and its histogram.

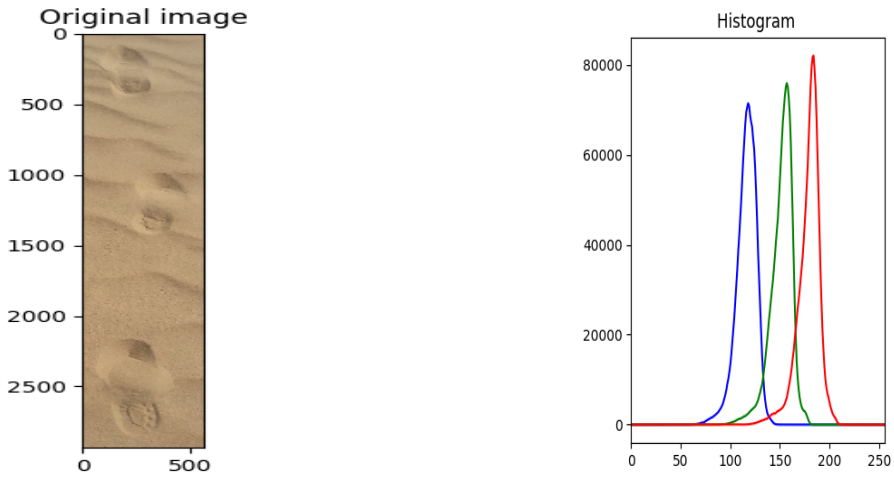


Figure 3.8: Image RGB color of foot steps trace and its histogram.

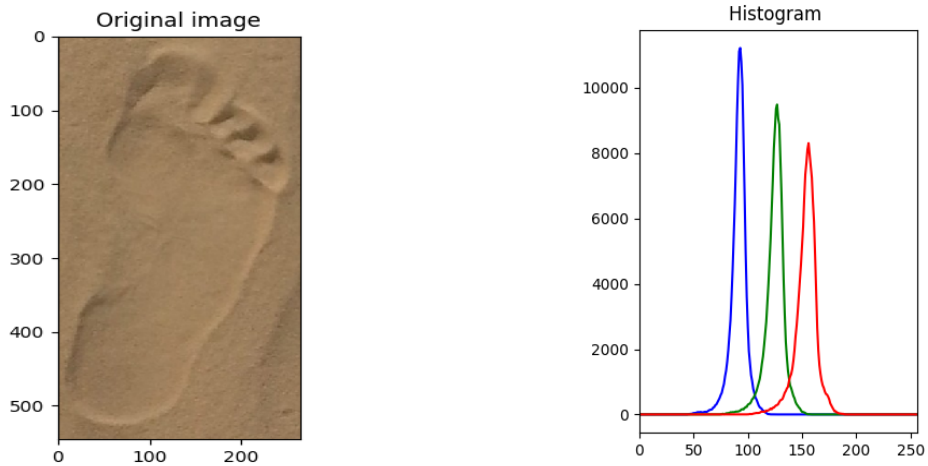


Figure 3.9: Image RGB color of barefoot trace and its histogram.

We can notice in these figures of two images examples, that each histogram has a part (50 - 220) of distribution the intensity values that are very high and it differs and not match in the same part of the intensity values, e.i. the color image is distributed in a larger part of the intensity values.

3.1.2.3.2 GrayScale enhancing

Usually, the image color information does not help to identify important edges or other features, i.e. the color information could be useless (noise). This type of enhancing reduces the complexity of the images and increases the accuracy of the methods used. One reason for converting an RGB image to GrayScale format in image processing is to get compression, i.e. the space of saving RGB image is very high then GrayScale image. Converting by using the Weighted (luminosity) method which applies the below equation [86]:
New grayscale image $= ((0.3 * R) + (0.59 * G) + (0.11 * B))$.

The histogram graphically of the grayscale image displays a range of 256 values, that are showing the distribution of pixel intensity values amongst those grayscale intensity values [87]. Figures 3.10, 3.11, 3.12 are showing each image with its histogram.

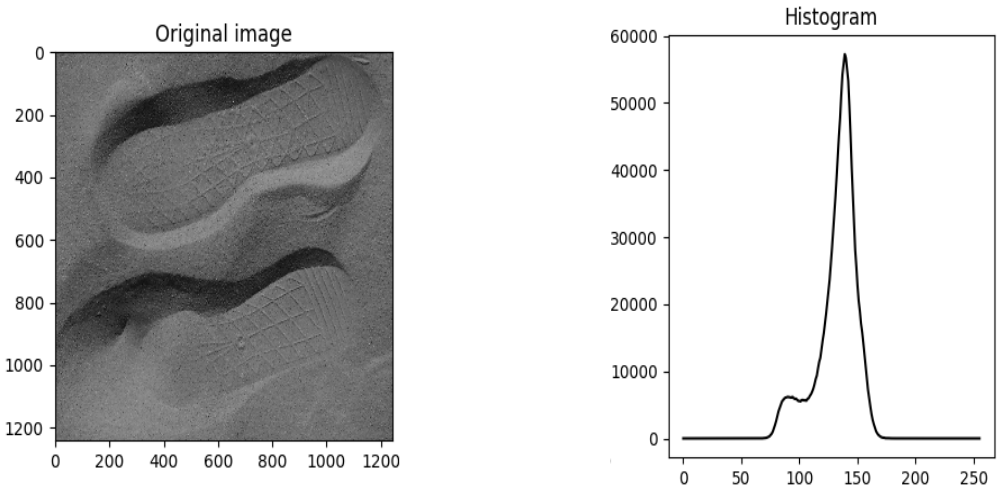


Figure 3.10: Grayscale of a balanced image with its histogram.

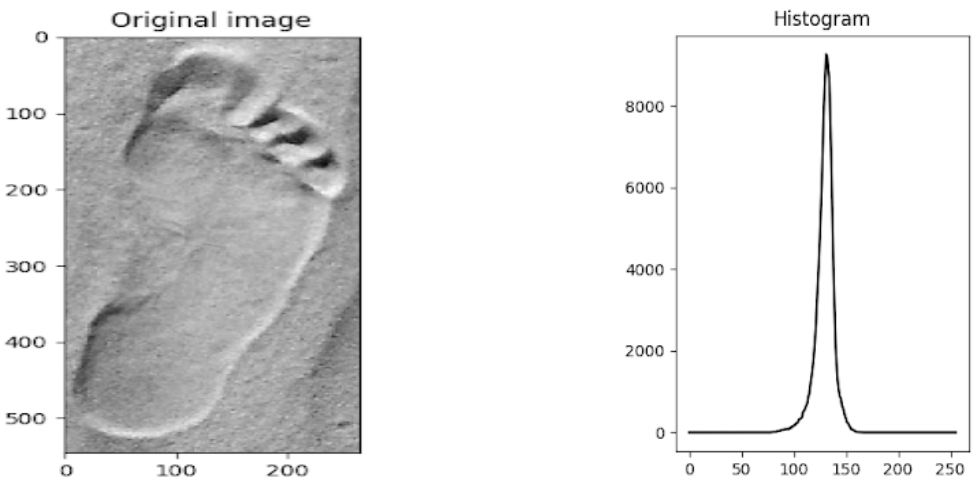


Figure 3.11: Grayscale of lighted image with its histogram.

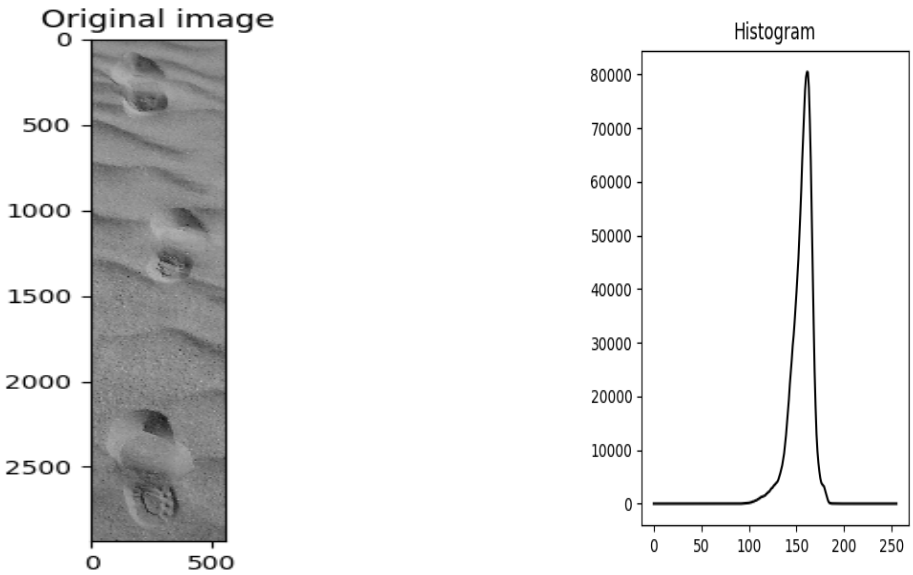


Figure 3.12: Grayscale of darkened image with its histogram.

These figures analyze grayscale HFT images using a histogram of the pixel intensity values. Through these histograms, we can see the distribution of the pixels of images with different intensity values. For each one, we distinguish different intensity values in each one. We note in the three images of the three figures examples, that each histogram has a part (125 - 150) of the intensity values that are very high, but it is very low in the rest part (0 - 125) and (150 - 255), e.i. This type of image is more difficult to enhance because it shows a low contrast of foot traces.

Figure 3.13 shows the histograms of three axes RGB for a RGB color image. So, we can see very clearly the difference between RGB color image and gray image. Through figure 3.13 we can see the same part of the intensity values for the three axes RGB (125 - 150), e.i. gray images of the pixel intensity values are very close. In this case of a data image, the **RGB color image is better than the grayscale image**.

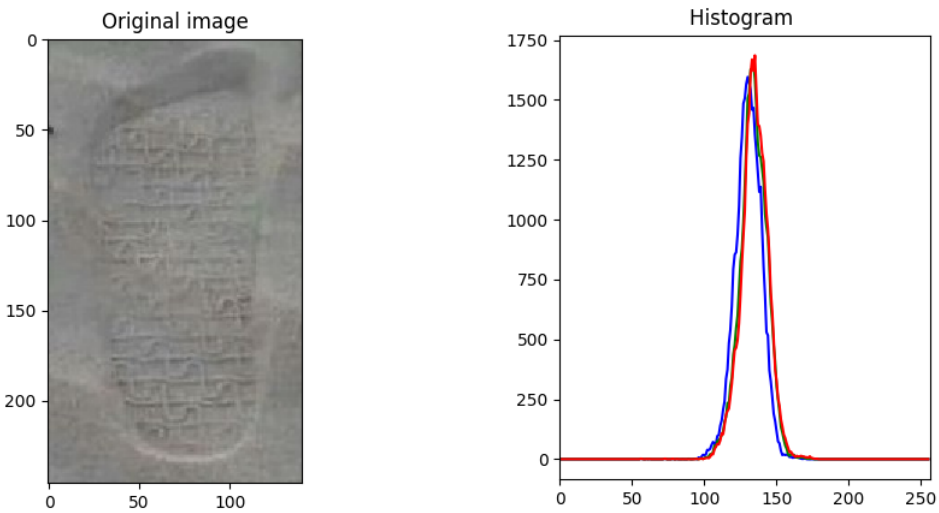
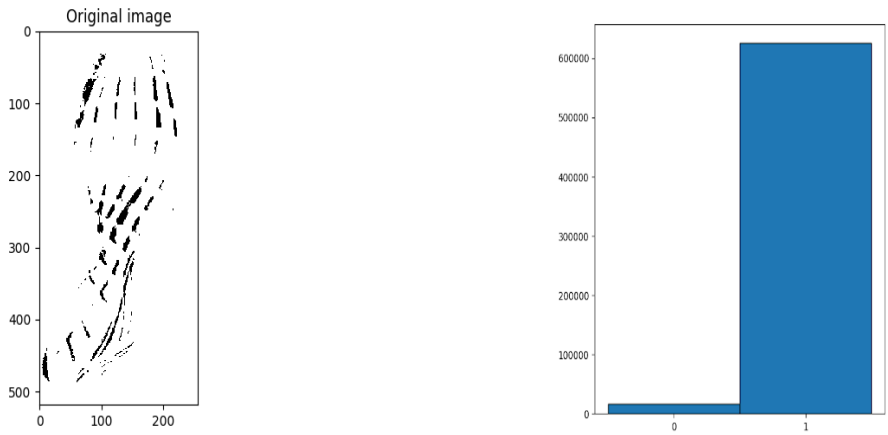


Figure 3.13: Color image-3 with its histogram.

Therefore, it is so difficult to apply the filters on grayscale images later. When we apply many enhancement methods, it hides most features and edges of HFT, which make it difficult to detect. Therefore, we exclude this case of improvements, it does not provide any image enhancement. Rather, it loses some of its essential features due to the pixel intensity values being very close.

3.1.2.3.3 Binary enhancing

A binary image is used as input to the feature extraction process and has an important role in generating unique features to distinguish several classes in image recognition. Based on a grayscale image, we get a binary image by using a Thresholding algorithm. Figure 3.14, 3.15 shows binary images with its histogram.



(a) (b)

Figure 3.14: Binary foot image and its histogram.

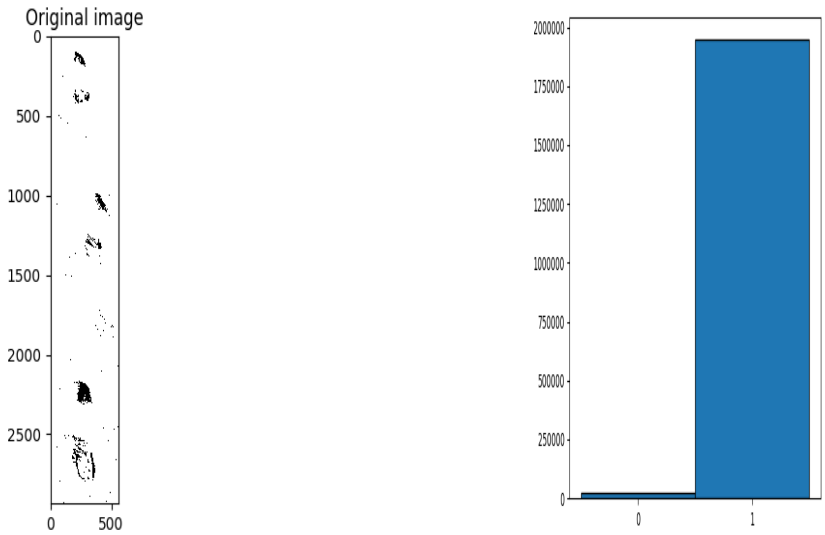


Figure 3.15: Binary foot steps image and its histogram.

This case is useless because it reduces the resolution of the image, and thus the image loses its important features. This leads to bad processing later. So, we rule out this case. And we choose the RGB color image for our process.

3.1.2.3.4 Contrast stretching

After we tried to improve the image by converting it into three types (RGB color, grayscale, binary), We found that the RGB color is the most appropriate for these data images. In this part, we will see with the contrast stretching which is applied directly to the RGB color image by modifying each pixel value. Contrast Stretching is one of the operations applied to the images, which improves the contrast of each pixel of the image [87]. The following images in figures 3.16, 3.17, 3.18 show the difference between the color images before and after applied contrast stretching.

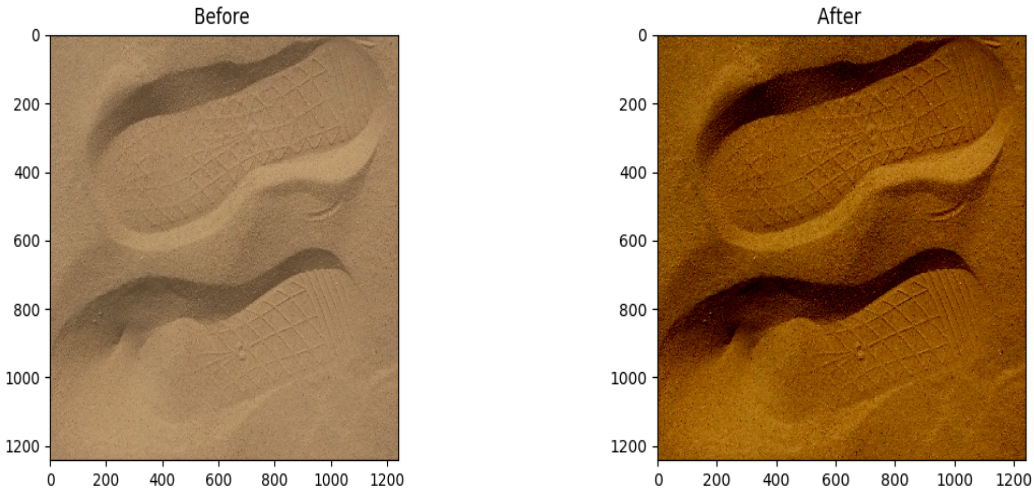


Figure 3.16: image-1 Before and after applying the contrast stretching.

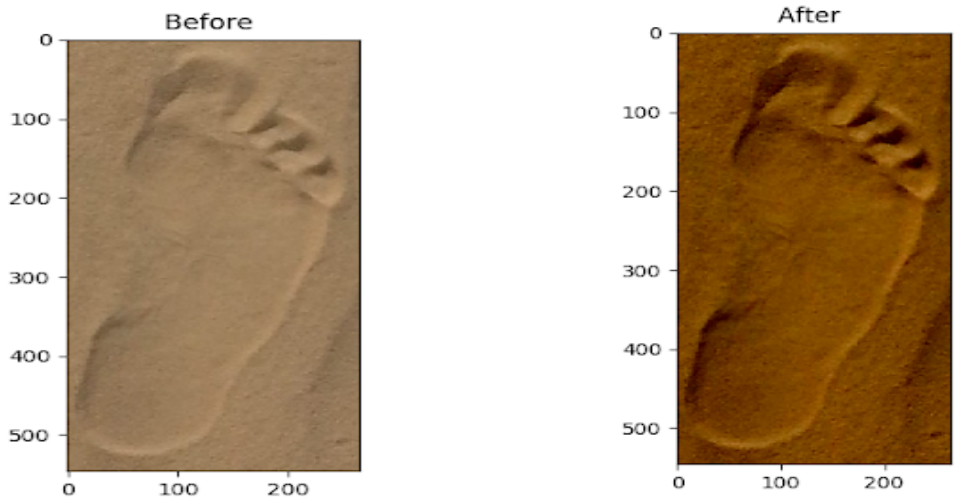


Figure 3.17: image-2 Before and after applying the contrast Stretching.

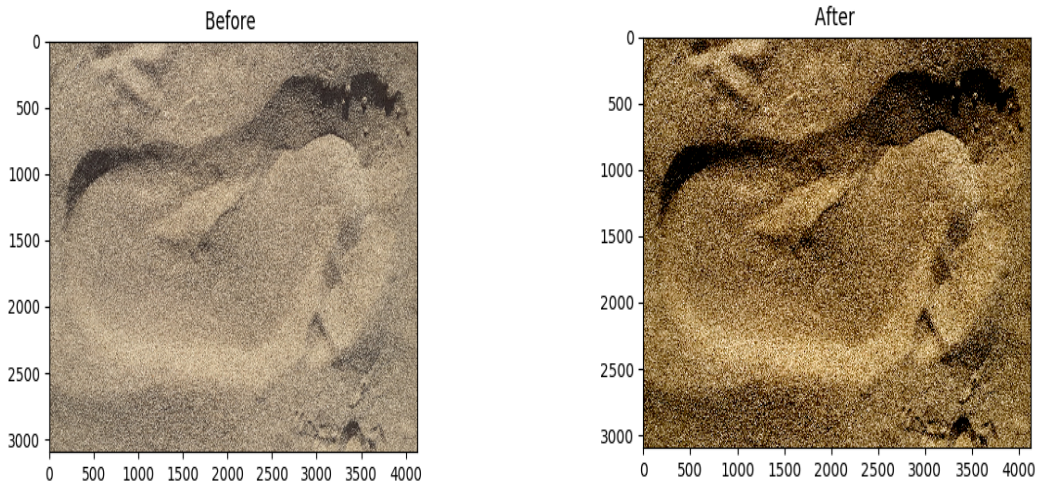


Figure 3.18: image-3 Before and after applying the contrast stretching.

3.1.3 Dataset constructing

In this step, we decided to propose two different constructions of a dataset(s) for two different systems. First, we analyze our processed data statistics for doing the primitive construction of the dataset. In this case, the term construction means classifying data images (labeling) according to the requirements of the research. Then, we propose two different distribution of the data images according to these systems:

- The first system: **One Model System (OMS)**.
- The second system: **Multi-Model system (MMS)**.

3.1.3.1 Primitive construction

In this part, we present the processed data statistics, and their construction according to our proposed systems, i.e. the construction of the first dataset for one model system, and the construction of the second dataset for the multi-models system.

After the preprocessing step, we get 55 individuals (volunteers) data. For each individual, we create a folder named with the person ID, i.e. classifying 55 labels identified by ID for each person. Each person has a folder that contains his images and a text file of the necessary information. In addition, we add additional information to the Excel file that contains all the personal information related to the data images for each person, the additional information (footwear, barefoot, age groupe, feet direction). Table 3.1 shows data statistics of different groups (sub-classes) of images.

characteristic	Type	Number of individuals
Barefoot / Footwear	Barefoot	22 (40.00%)
	Footwear	38 (69.09%)
	Barefoot & Footwear	05 (09.09%)
Age Groupe	Kids	15 (27.27%)
	Teenagers	19 (34.54%)
	Adults	21 (38.18%)
Feet direction	Straight	27 (49.09%)
	In-Toeing	09 (16.36%)
	Out-Toeing	19 (34.54%)

Table 3.1: Processed data statistics for 55 individuals.

Through table 3.1, we see 3 principal features barefoot/footwear, age groups, feet direction. After observing the features and evaluating them for 55 volunteers, we finish analyzing each feature in the following points:

- For the barefoot / footwear feature, we have:
  - → 44% of total participants who have barefoot images.
  - → 69.09% of total participants who have footwear images.
  - → 9.09% of total participants who have barefoot & footwear images.
- In the age groups feature, we have:
  - → 27.27% of the total participants are Kids ([5, 12] years old).
  - → 34.54% of the total participants are Teenagers ([13, 18] years old).
  - → 38.18% of the total participants are Adults ([19, <=] years old).
- Finally, feet direction feature, we have:
  - → 49.09% of total participants who have Straight feet.
  - → 16.36% of total participants who have In-Toeing feet.
  - → 34.54% of total participants who have Out-Toeing feet.

3.1.3.2 Dataset construction

The primitive dataset contains representative samples of HFT images. The construction is a labeling process, which distributes the primitive dataset images into classes (labels) according to the desired model task of classification. In a model, the dataset is the important element in the process of building the model, i.e. the dataset used in the training and evaluates a model. In this step, we constructed the datasets from the primitive dataset. Ultimately, we summarize in a table the data set construction process.

3.1.3.2.1 Constructing one model dataset

The proposed system (one model) is constructed from the only model. In this part, we create one dataset of 24 labels. Figure 3.19 shows dataset representation in a hierarchy form (tree structure), where the ends of the tree (tree leaves) present labels.

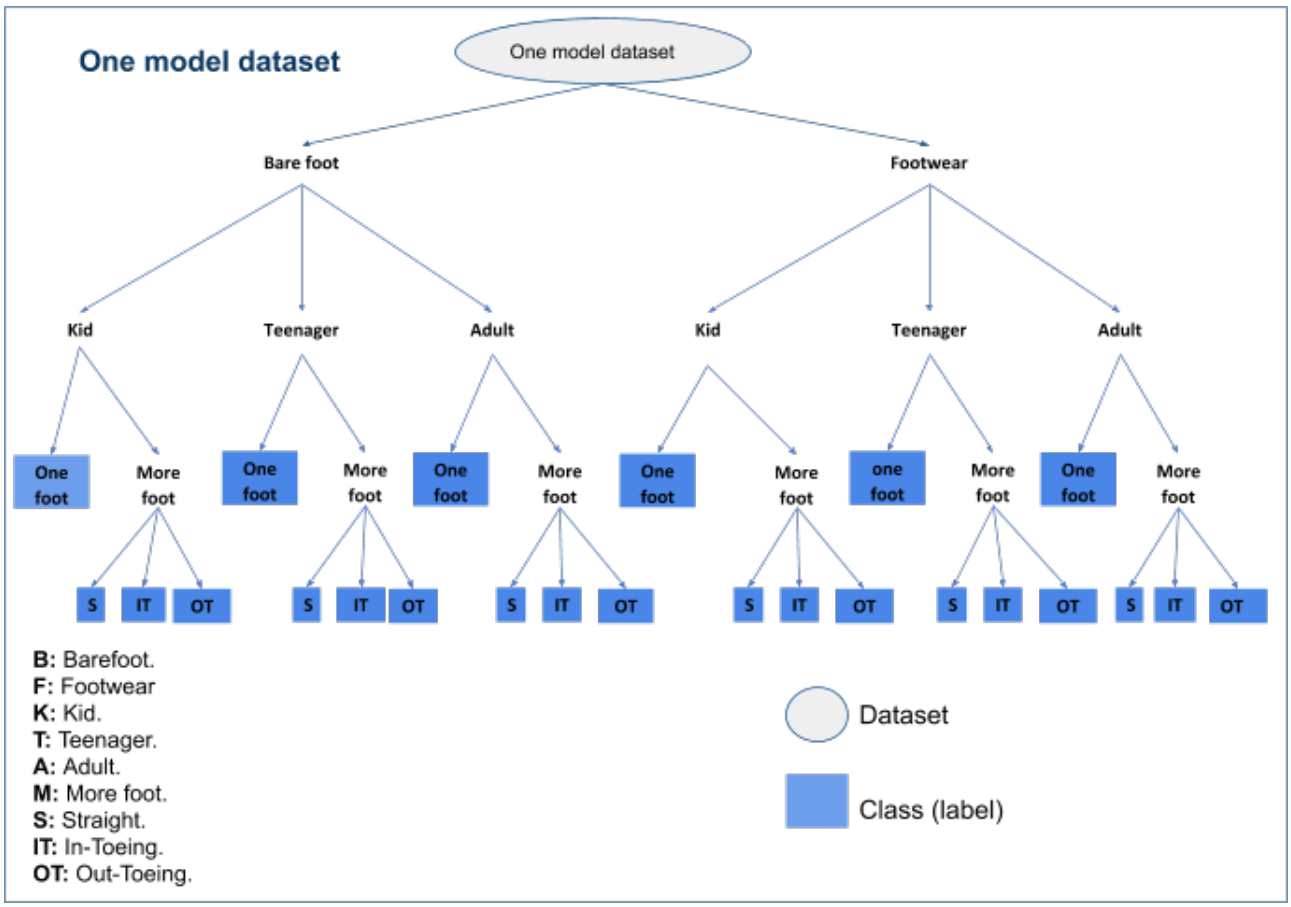


Figure 3.19: Tree structure of dataset for OMS.

Finally, we get one dataset of 24 classes (labels) are listed below:

- L1: Footwear trace, for a kid person, includes one foot.
- L2: Footwear trace, for a kid person, includes more foot, Straight type feet direction.
- L3: Footwear trace, for a kid person, include more foot, in-toeing type feet direction.
- L4: Footwear trace, for a kid person, include more foot, out-toeing type feet direction.
- L5: Footwear trace, for a teenager person, includes one foot.
- L6: Footwear trace, for a teenager person, includes more foot, Straight type feet direction.
- L7: Footwear trace, for a teenager person, include more foot, in-toeing type feet direction.
- L8: Footwear trace, for a teenager person, include more foot, out-toeing type feet direction.
- L9: Footwear trace, for an adult person, includes one foot.
- L10: Footwear trace, for an adult person, includes more foot, Straight type feet direction.

- L11: Footwear trace, for an adult person, includes more foot, in-toeing type feet direction.
- L12: Footwear trace, for an adult person, includes more foot, out-toeing type feet direction.
- L13: Barefoot trace, for a kid person, includes one foot.
- L14: Barefoot trace, for a kid person, includes more foot, Straight type feet direction.
- L15: Barefoot trace, for a kid person, include more foot, in-toeing type feet direction.
- L16: Barefoot trace, for a kid person, include more foot, out-toeing type feet direction.
- L17: Barefoot trace, for a teenager person, includes one foot.
- L18: Barefoot trace, for a teenager person, includes more foot, Straight type feet direction.
- L19: Barefoot trace, for a teenager person, includes more foot, in-toeing type feet direction.
- L20: Barefoot trace, for a teenager person, includes more foot, out-toeing type feet direction.
- L21: Barefoot trace, for an adult person, includes one foot.
- L22: Barefoot trace, for an adult person, includes more foot, Straight type feet direction.
- L23: Barefoot trace, for an adult person, includes more foot, in-toeing type feet direction.
- L24: Barefoot trace, for an adult person, includes more foot, out-toeing type feet direction.

3.1.3.2.2 Constructing multi-models dataset

The proposed muli-models system is constructed from more than one model. Each one created for a specific task of classification in this system. According to the proposed models and their tasks, we implemented the first tree structure and then improve (optimization). Before optimization, we create 15 datasets. Figure 3.20 shows more about these datasets through the representation in the same structure tree (figure 3.19), where the nodes present the datasets and the tree leaves present its labels.

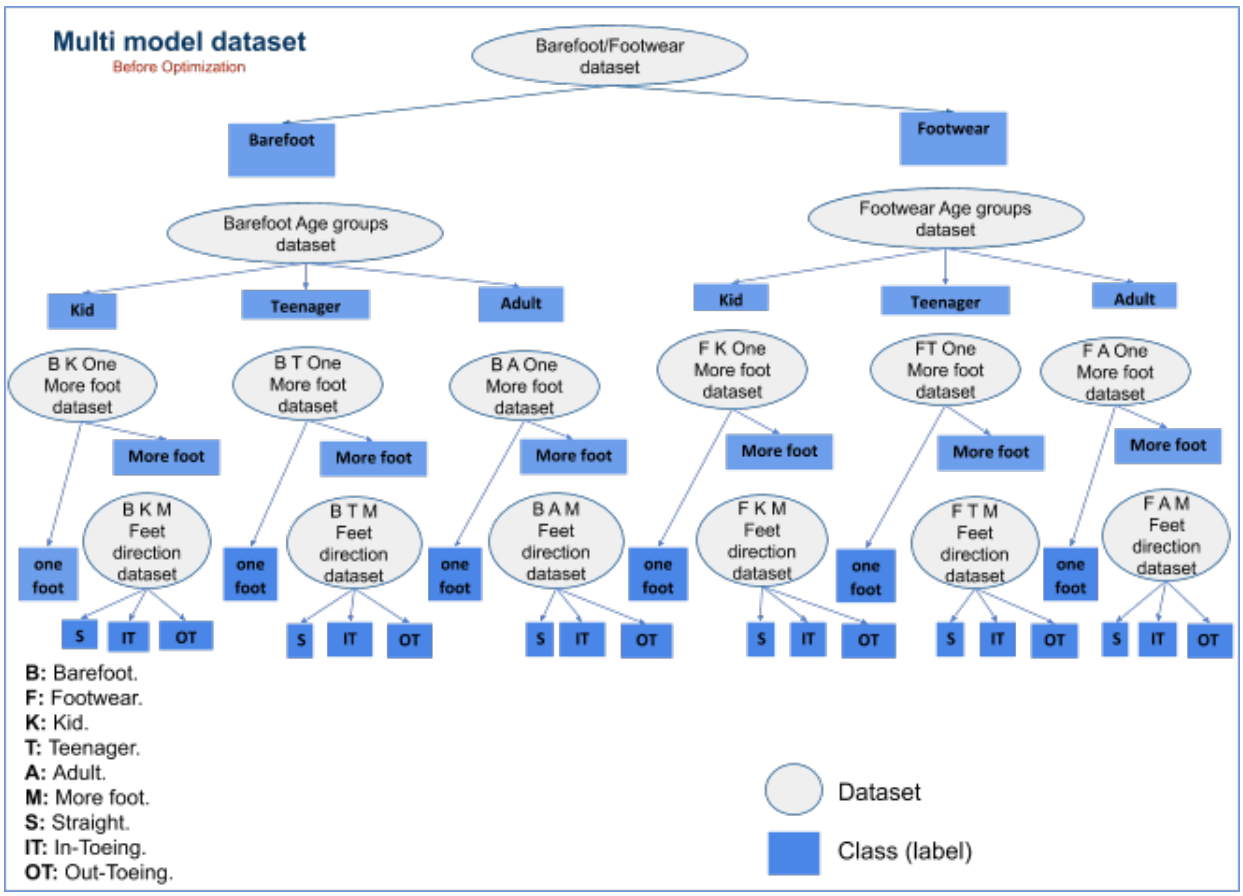


Figure 3.20: Tree structure of dataset for MMS before optimization.

After a set of improvements applied to the previous tree structure, we optimized this structure to be better and eliminate redundancy. Also, to avoid some states such as too little size of the dataset leads to inaccurately training a model, an imbalance in the image distributions in labels. After optimization, we have 9 datasets as shown in figure 3.21.

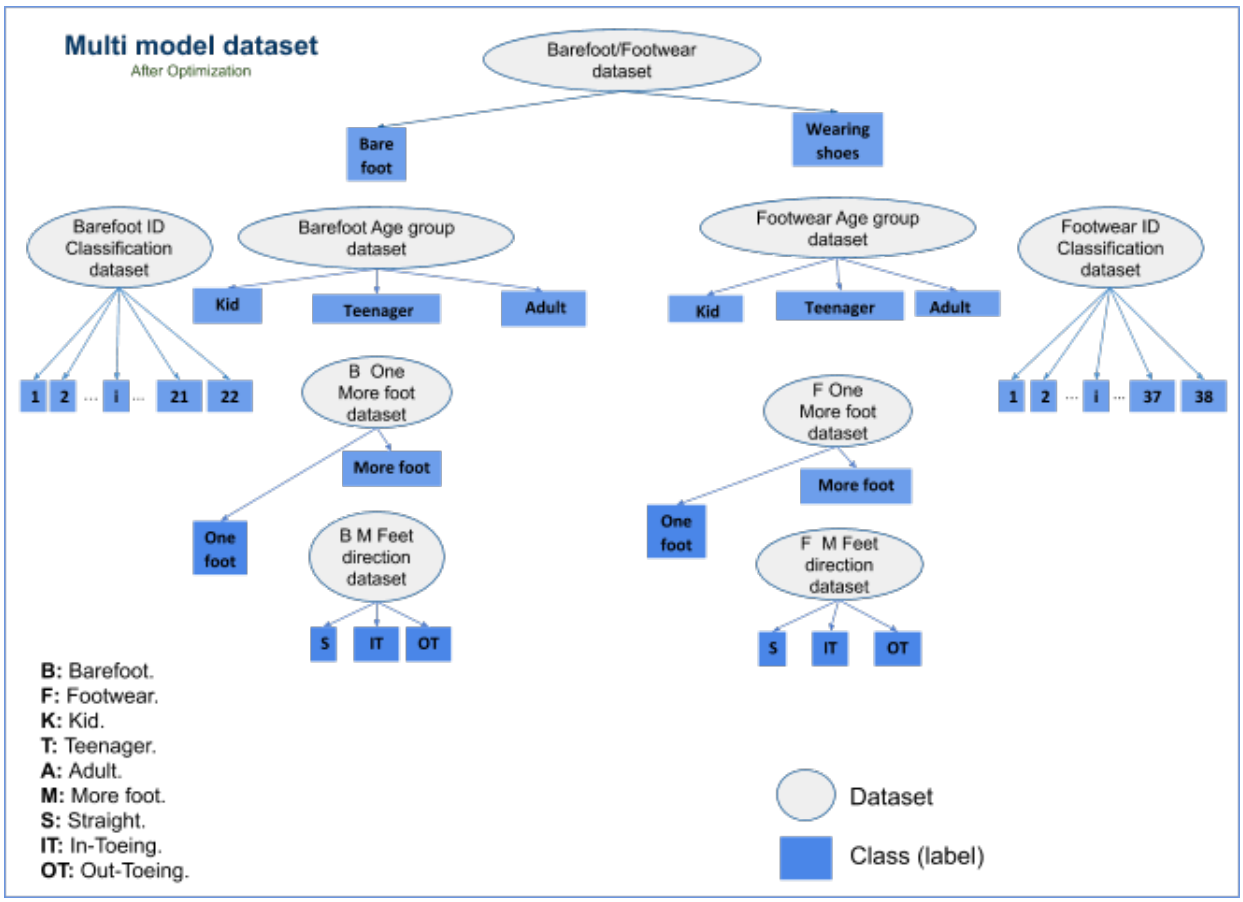


Figure 3.21: Dataset classes structure for MMS after optimization

Through the above figure, there are nine different datasets, we explain each one separately:

- **Barefoot / Footwear dataset**
  We split the primitive dataset into two subsets, the first one contains foot traces images of all person footwear, i.e. persons wear shoes. The second subset includes foot traces images of all persons barefoot, i.e. persons without shoes. There is a possibility that the same person belongs to both subsets.

- **Age group dataset**
  Based on the precedent dataset 'Barefoot / Footwear', we take each class ('Footwear' and 'Barefoot') to create two new datasets. The first one is the "Barefoot Age group" dataset, and the second is the "Footwear Age groups" dataset. In each dataset, we split it into three classes, Kids, Teenagers, and Adults. So, we have now two new datasets, each one has three labels.

- **One / More foot datasets**
  Based on the precedent dataset 'Barefoot / Footwear', we take each class ('Footwear' and 'Barefoot') to create two new datasets. The first one is the "Barefoot One / More foot" dataset and the second is the "Footwear One / More foot" dataset. In each dataset, we split it into two classes, One foot and More foot. So, we have now two new datasets, each one has two labels. In each set, we split it into two classes, One foot and More foot. So, we have now two datasets, each one has two labels.

- **Feet direction dataset**
  Through the precedent datasets 'One / More foot', which presents two datasets 'One More foot Footwear' and 'One More foot Barefoot'. We take one class of each dataset, which are 'More foot for Footwear' and 'More foot for Barefoot' to create two new datasets. The first one is the 'Barefoot Feet direction' dataset and the second is the 'Footwear Feet direction' dataset. In each selected dataset, we split it into three classes Straight, In-Toeing, and Out-Toeing. So, we also have two new datasets, each one has three labels.

- **ID classification dataset**
  Based on the precedent dataset 'Barefoot / Footwear', we take each class ('Footwear' and 'Barefoot') to create two new datasets. The first one is the "Barefoot ID classification" dataset and the second is the "Footwear ID classification" dataset. In the first dataset, we split it into 22 classes (1, 2, ...,22), and the second set, we split it into 38 classes (1, 2, ...,38). Where the classes (1, 2, ...,22) represent the number of people belonging to a class 'Barefoot', and the classes (1, 2, ...,38) represent the number of people belonging to a class 'Footwear'. So, we have now two datasets, the first one has 22 labels and the second has 38 labels.

3.1.3.2.3 Dataset construction summarized

The total datasets size is 673 images for 55 individuals. We summarize the dataset(s) construction for each system in Table 3.2 that shows the datasets for each system with the class size of each one.

System	ID	Datasets	Labels (classes)	Dataset size
One model	1	One-model dataset	[1..24]	673
Multi-model	2	Barefoot / Footwear	Barefoot	261
			Footwear	457
	3	Barefoot Age group	Kid	108
			Teenager	31
			Adult	109
	4	Footwear Age group	Kid	30
			Teenager	230
			Adult	214
	5	Barefoot One / More foot	One foot	82
			More foot	166
	6	Footwear One / More foot	One foot	158
			More foot	236
	7	Barefoot Feet direction	Straight	73
			In-toeing	42
			Out-toeing	49
	8	Footwear Feet direction	Straight	155
			In-toeing	57
			Out-toeing	96
	9	Barefoot ID classification	[1..22]	216
	10	Footwear ID classification	[1..38]	457

Table 3.2: All dataset with its total size.

In order for the above data sets to be more clear and to have a more comprehensive idea and better visualization of their relation to each other, their number, proportion, and statistical comparison. We assign all the data set values in one column chart which is as the following chart 3.22 :

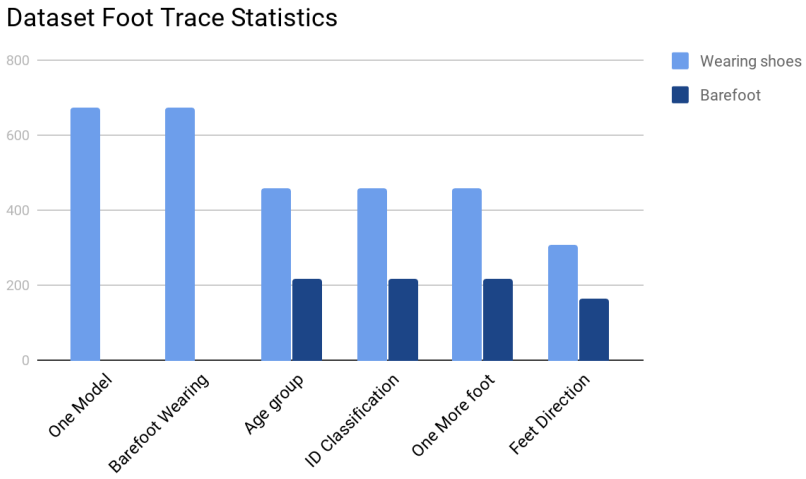


Figure 3.22: Char 1: Dataset human foot trace statistics.

Through this chart column, we can distinguish that:

- the datasets that include the other datasets are:
  - Barefoot/Footwear.
  - One Model datasets.
- A dataset that is equal to another dataset:
  - Barefoot/Footwear = One Model.
  - Age Group = ID Classification = One/More foot.

3.1.4 Dataset preparing

Images in the dataset had different sizes and its label is not binarized, ...etc. Therefore, the images have been processed to use by the model for training, feature extraction, classification,...etc. The same processing tasks applied to all image systems to make it match the arguments of the target model(s). The most important sides in the processing of the input image for preparing it,

- Convert an image to an array.
- Swapping image color channels to RGB.
- Resizing images to behave a fixed size as input tensor.
- Normalizing image.
- Binarizing labels.

3.1.4.1 Convert an image to an array

To make the model read each pixel of the image. For this, we have to convert the image into a two-dimensional matrix. This matrix is a set of elements, each one is represented by three RGB color values. The values indicate the intensity of red, green, and blue of the pixel. The value of each color is between 0 and 255.

3.1.4.2 Convert the image from BGR to RGB color format

To preserve the original color format of the image, we return the values to the RGB color format by inverting the three values for each pixel from BGR color format to RGB color format, i.e. from (blue, green, red) to (red, green, blue).

When reading an image by some specific functions in order to convert it into an array, we get an array of values of the BGR color format although the original color format of the image is RGB. This auto-inversion of the color format from RGB to BGR while reading the image due to some of the functions from specific libraries used for reading the image, such as CV2.imread(image) from library OpenCV in Python3.

3.1.4.3 Resize image

Makes the size of the image input equal to the size of the model input layer. In our system, we tried to work on many values such as 250, 299, 320, 350, 400,... etc. But we fixed our choice on two values:

- **299x299**: it applies to all models except two models (feet direction models).
- **350x200**: it applies to only two models (feet direction models) because applying resize 299x299, it will lose the principal features of the image.

3.1.4.4 Normalization image

Makes the scale values for each pixel into the range 0-1, normalization is used to make model training less sensitive to the scale of features. This allows our model to converge to better weights and, in turn, leads to a more accurate model [87].

3.2 Systems models construction

In this phase, we focus on building the systems model (s). These models consist of two parts, a feature extractor and a classifier. The feature extractor is chosen from the five proposed pre-trained models. The classifier was chosen as the best of the 3 types of classifiers. Based on the results of prediction models, the matching function introduces a suitable person where the making decision step decides that person is known or unknown.

3.2.1 Feature extractor

In this step, we focus on the representation of the proposed HFTs scheme. Figure 3.23 shows the general representation of HFTs, i.e. the main parts common to both systems, OMS and MMS. Firstly, we present a simple scheme for each system. We concluded this step with a comparative experiment for selecting the best model for extracting the features.

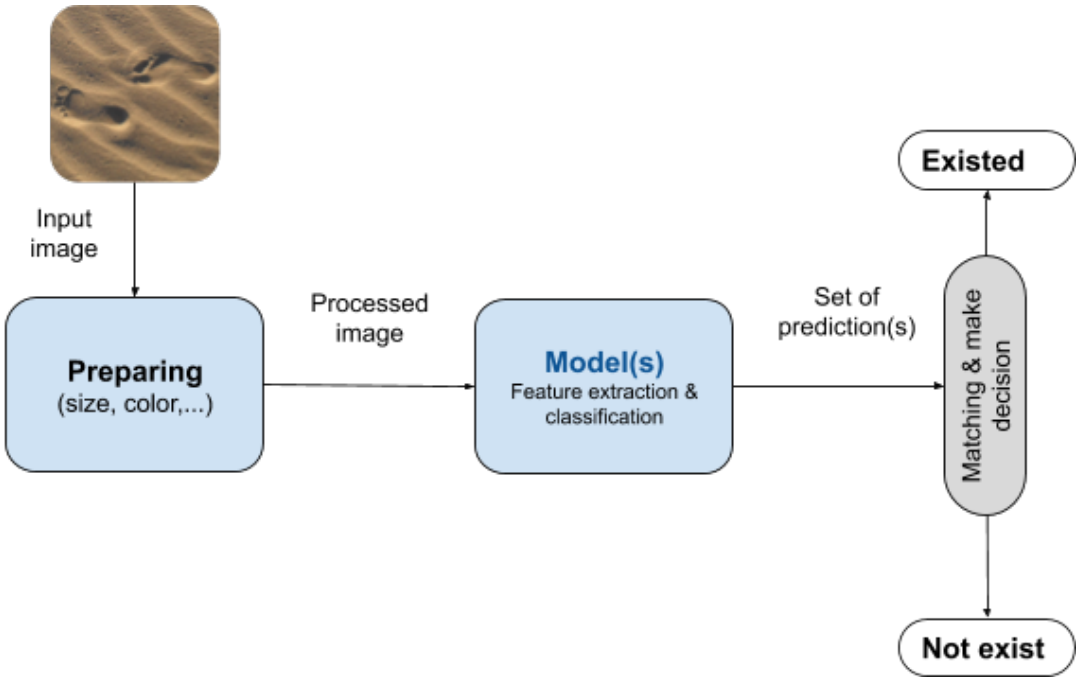


Figure 3.23: General HFT system scheme.

3.2.1.1 HFT systems schema

In this part, we present two proposed HFTs schema. The first system is based on the one model approach that has one model and the second system is based on the multi-models approach that has nine models.

3.2.1.1.1 One model system

Figure 3.24 shows the HFTs schema of the one model approach, which illustrates the main parts of working the system, dataset preparing, CNNs features extractor, classifier, matching, and making decisions.

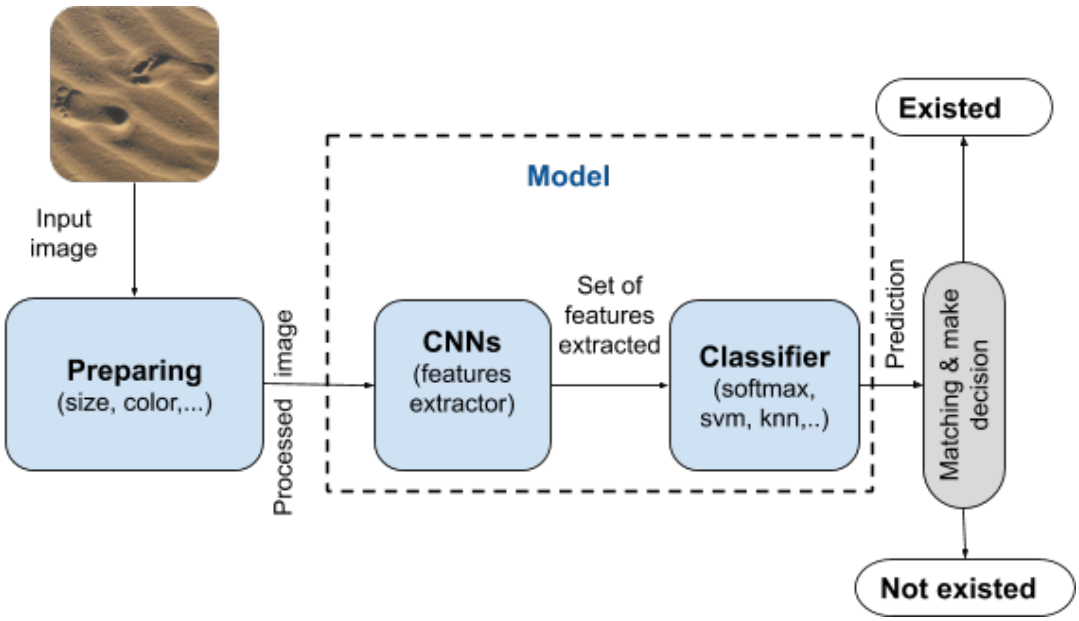


Figure 3.24: HFT of One model system schema.

In the above schema, we notice that this approach includes only one model which consists of CNNs and a classifier, where CNNs for extracting the image feature, and a classifier for performing the task of classification and giving a prediction result.

3.2.1.1.2 Multi-model system schema

Figure 3.25 shows the HFTs schema for the multi-models approach, which illustrates the main parts of the system, dataset preparing, CNNs features extractor, classifier, matching, and making decisions.

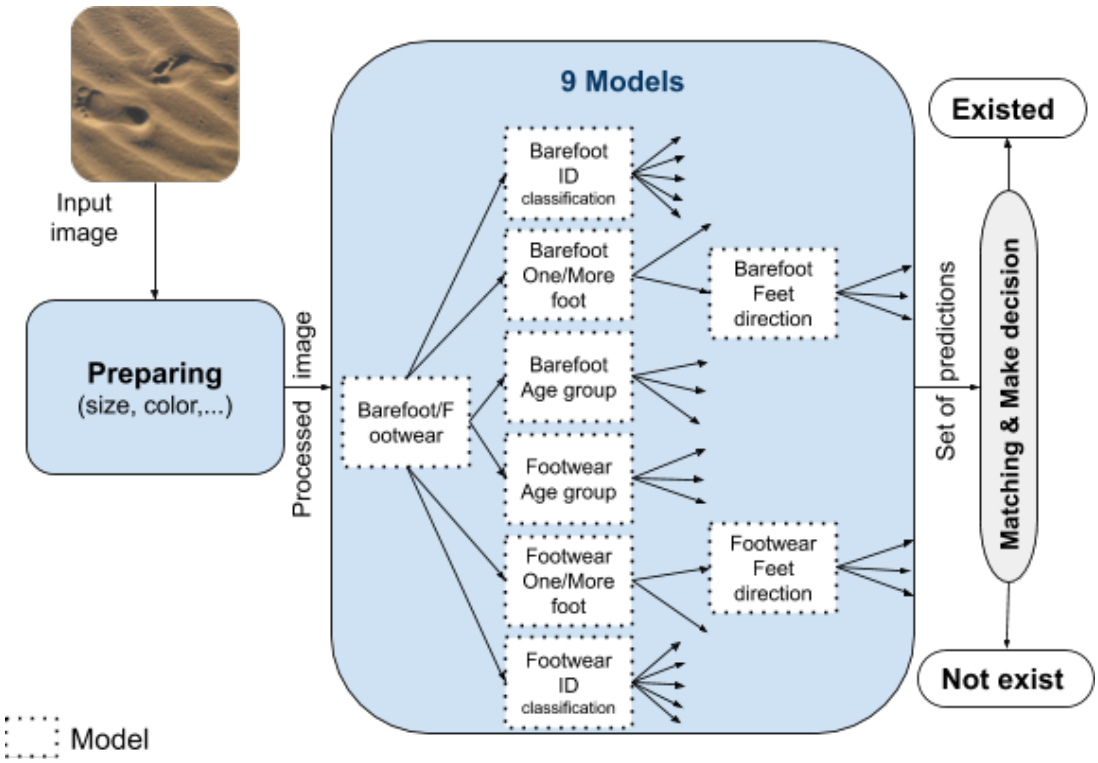


Figure 3.25: HFT of the multi-models system schema.

In the previous sceme, we notice that this approach includes nine models (9 models). Each model consists of CNNs and a classifier, where CNNs for extracting the image feature, and a classifier for performing the task of classification and giving a prediction result.

It is to note that the process of appling these models is in the form of a hierarchical taxonomy, i.e. the determined model to be applied later, depend on the prediction result of the previous model. So, in each process, the system uses just 4 models (if the prediction result of One/More model is one model), or 5 models (if the prediction result of One/More model is more model), not all models (9 models).

3.2.1.1.3 Difference between OMS & MMS schema

According to these previous schemes, we notice almost both of them have the same parts. The principal differences between these systems are in the mechanism of application of models (hierarchical taxonomy) and the number of the used models, i.e. the number of predictions results.

3.2.1.2 Comparative experiment to selecting features extractor

In this part, we decided to use the CNNs model as a features extractor. For this, we have conducted a comparative experiment on 6 models in terms of accuracy. The 6 models are one Basic model which was proposed by us and 5 pretrained models which are VGG16, VGG19, ResNet50, Inception V3, Xception.we use the softmax layer as classifer. The comparative experiment is conducted for choosing the best one for extracting the image features and the best performance with our dataset type. The selection result based on

the classification accuracy of these models for the chosen dataset "Barefoot/Footwear dataset". The experiment results are shown in the table 3.3.

- **Parameters configuration**:
  - Training data: 80%
  - Validation data: 20%
  - Testing data: 20%
  - Initial learning rate (INIT_LR):$1e^{-}3$)
  - Batch size (BS) : 10
  - Epochs: 70
  - Number of classes: 2 ("barefoot","footwear")
  - Dataset size: 673 images

Features extractor model	Input size	Data augmentation	Accuracy
Basic proposed model	250*250*3	Yes	**81,94**
		No	**79,86**
	350*350*3	Yes	**84.03**
		No	**75,00**
VGG16	224*224*3	Yes	**86,11**
		No	**85,42**
	350*350*3	Yes	**88,19**
		No	**88.89**
VGG19	299*299*3	Yes	**86,81**
		No	**87,50**
	350*350*3	Yes	**85,42**
		No	**90,28**
ResNet50	224*224*3	Yes	**67,78**
		No	**63,64**
	350*350*3	Yes	**67,63**
		No	**66,43**
Inception V3	299*299*3	Yes	**93,75**
		No	**94,44**
	350*350*3	Yes	**93.06**
		No	**90,28**
Xception	299*299*3	Yes	**95,83**
		No	**97,22**
	350*350*3	Yes	**96,53**
		No	**95,14**

Table 3.3: Results of a comparative experiments of different feature extractor models.

From the previous table, we notice that the accuracy value obtained from the Xception model is higher than the other accuracy value of models for the 4 cases (**95.83, 97.22, 96.53, 95.14**).Which has the highest value **(97.22%)** with **no augmentation data** and an input size image of **299x299**.

Based on the comparison results, we take the Xception model as the best model for the feature extraction step. So, we will use Xception in all other systems models to extract the features, with the parameters:

- input size: 299 x 299.
- without augmentation data.
- Training data: 80%
- Validation data: 20%
- Testing data: 20%
- Initial learning rate ($INIT_LR$): $1e^{-}3$
- Batch size (BS) : 10

The aim of using the softmax classifier in this experiment is only to choose the best model through the accuracy value. So, we focus only on the Xception model without the classifier layer, i.e. we just use the CNNs layers of the Xception model that is responsible for extracting features of an image by using its filters (convolution layers) without any modification in the weights. These CNNs part layers are called the freezing of the weights of the CNNs which are used for extracting the features only. Figure 3.26 shows the features extractor CNNs in the Xception model (freezing CNNs of Xception).

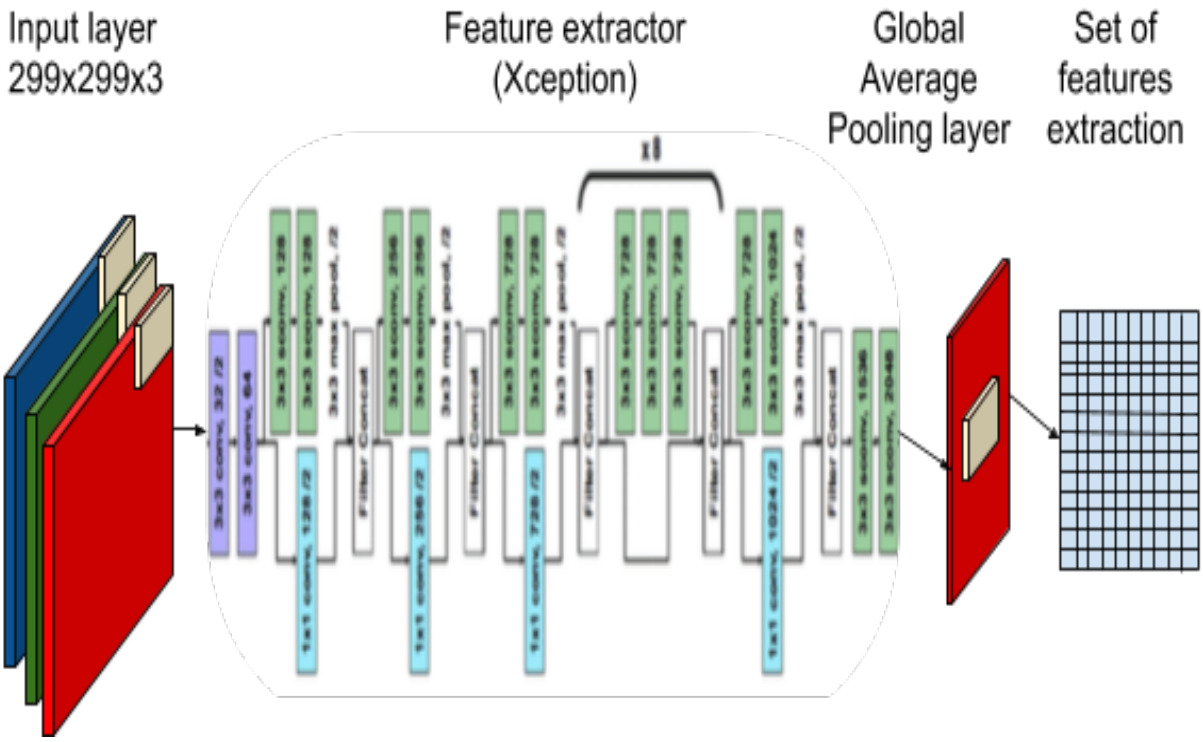


Figure 3.26: Features extractor in the Xception.

3.2.2 Classification

From the previous step, we see the HFT schema for each system which involves the Xception model for extracting the image features and a classifier for giving a prediction.

Previously, we focus on the feature extractor only. In this step, we focus on the classifier which has many types. For the Xception pretrained models, we proposed 2 methods of using the classifier:

1. We use an Xception pre-trained model for extracting features from the images dataset, and we removed the last layer of the Xception, which simply takes a probability for each of the 1000 classes in the ImageNet and replaces it with a layer that takes N probabilities, where N is the number of labels for a dataset. In addition, we freeze weights of every layer in the Xception model so that they do not train, we want our feature extractor to stay as before. Then, relating this freezing layer (untrainable layer) with the softmax layer (trainable) for classification. After that, we applied the training on the produced CNNs architecture(freezing layer with softmax layer) to train the classifier on the features extracted from the freezing layer. This method is called **transfer learning**.

2. We use an Xception pre-trained model as a feature extraction mechanism which returns a new structured data. This new dataset stores the features extracted of the image dataset. Then, it is used to train a linear classifier (SVM, KNN, ..). This method is called the **features extracted**.

Therefore, we conducted the final comparative experiment which applied to each of the 10 models (all models in both systems). Each model architecture consists of two parts, feature extractor layers in the Xception model and classifier as shown in figure 3.27 .

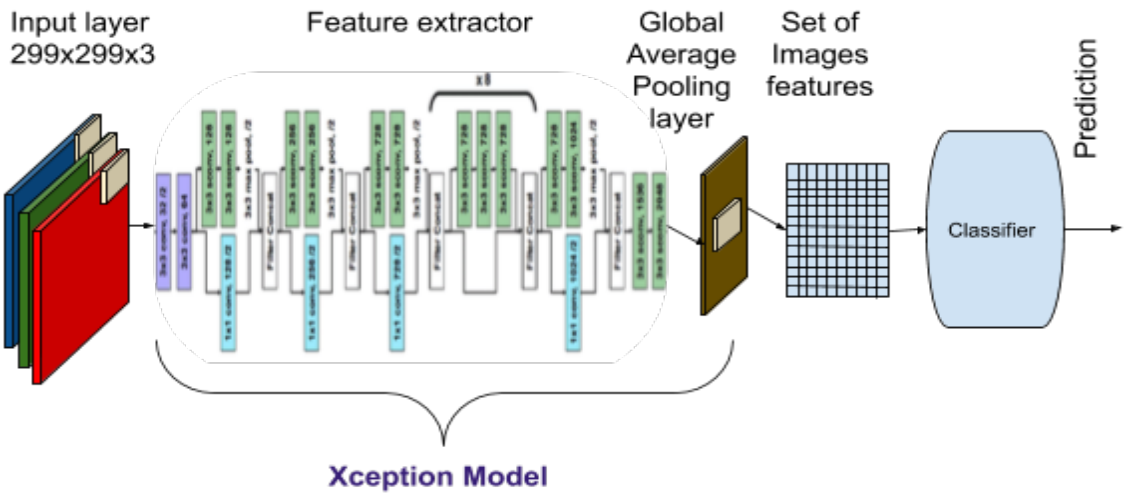


Figure 3.27: Model architecture.

From the previous architecture, we notice that the model consists of 2 parts. The first part represents a feature extractor layer in the Xception model (freezing weights). The second part represents a classifier (Softmax, SVM, KNN). We indicate that the same model is used in each system so the same model architecture.
In this experiment, we can distinct 3 cases of the model as shown in the figures 3.28, 3.29, 3.30

1. **Model case1**: Xception feature extractor with a Softmax classifier.

2. **Model case2**: Xception feature extractor with the SVM classifier.
3. **Model case3**: Xception feature extractor with the KNN classifier.

Lets first deduce the best classifier used in our study from the accuracy value. Hence, to deduce the best system from our proposed system, OMS or OMM.

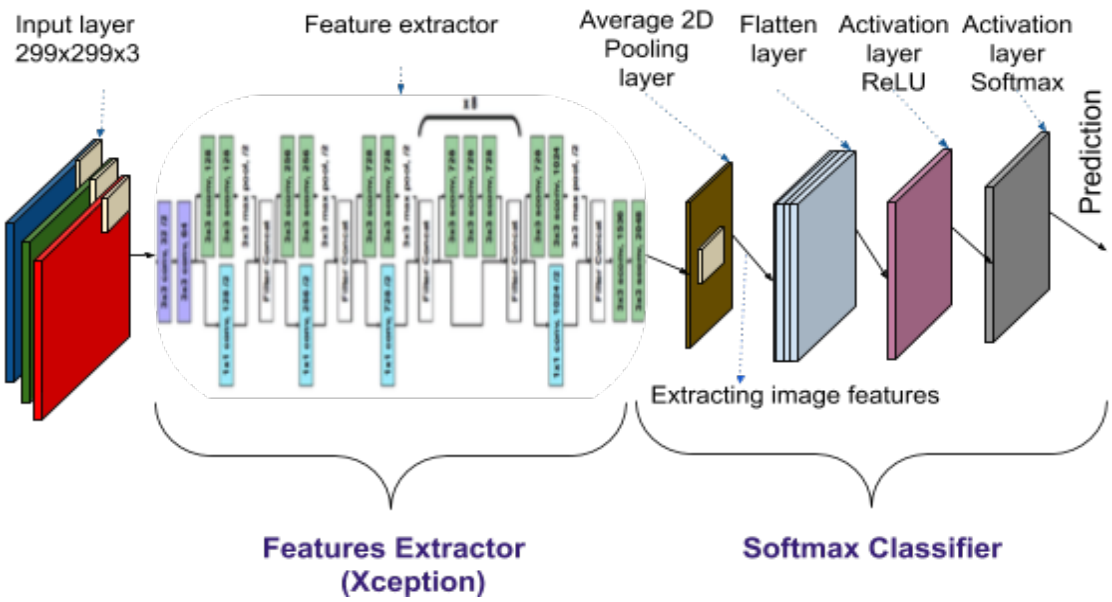


Figure 3.28: Model case1 (Softmax).

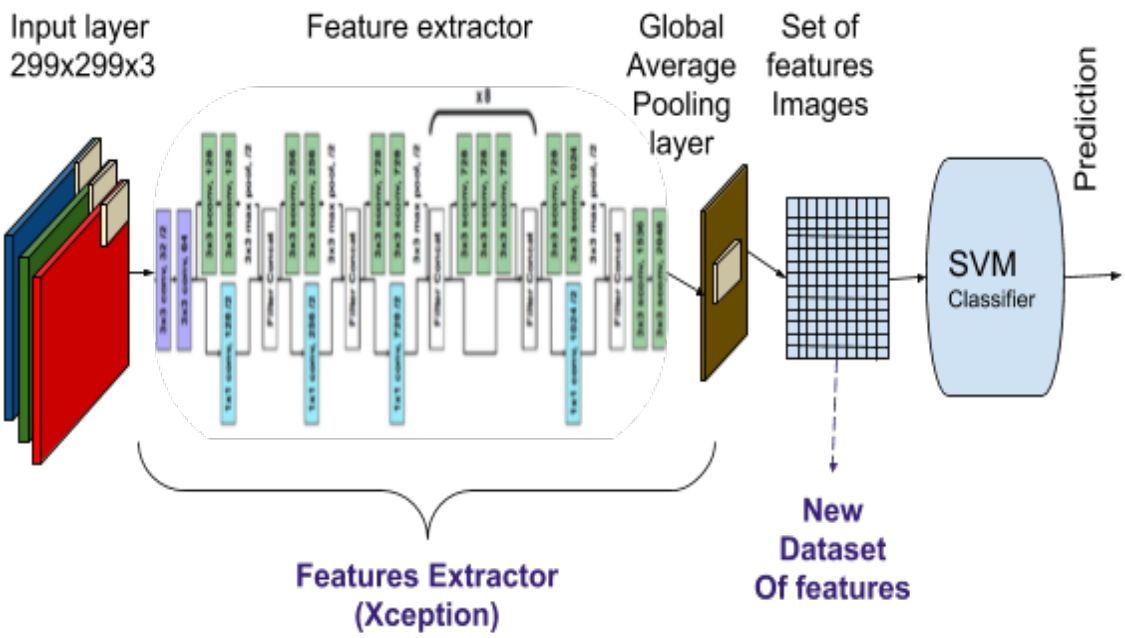


Figure 3.29: Model case2 (SVM).

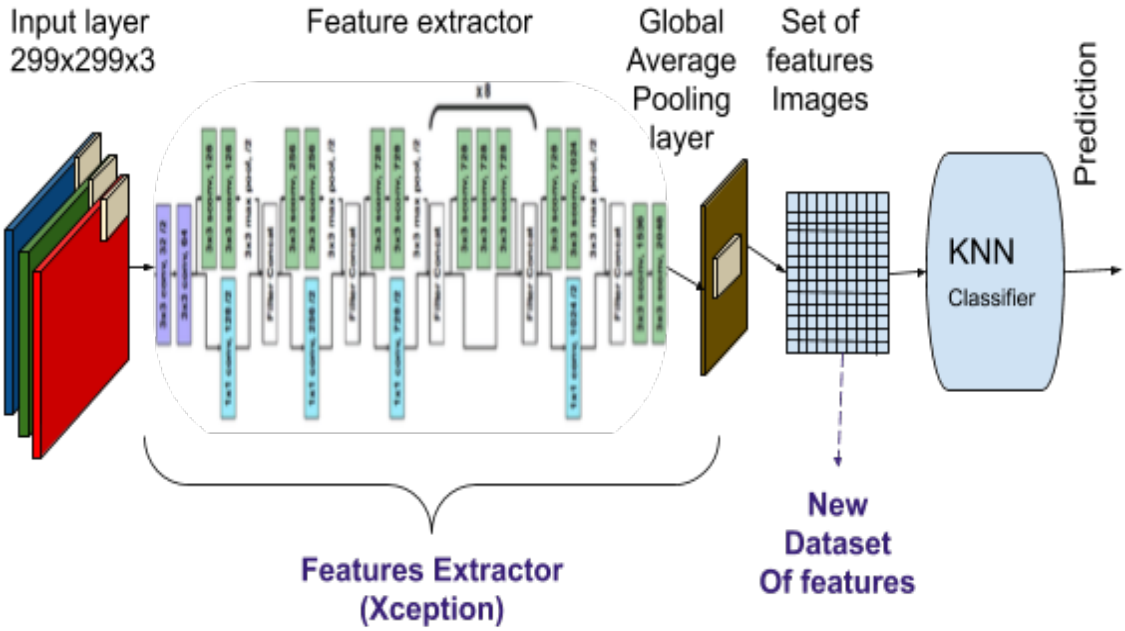


Figure 3.30: Model case3 (KNN).

3.2.3 Matching

The matching step is matching the obtained results with existing information. We mention the proposed matching algorithms in Algorithm 1 and Algorithm 2 to search for a congruence degree between the predicted information from the model(s) and information of a person who was recognized previously by the system. The matching algorithm is used in both systems which perform the same task, the difference just in the number of function arguments (matching algorithm). All the possible arguments for both systems:

- P1: Barefoot or Footwear.
- P2: Kid or Teenager or Adult.
- P3: One or More foot.
- P4: Straight or In-toeing or Out-toeing, (if the input image contains more foot trace, else P4: none).
- P5: Probability of prediction the One model.
- Top5: ID classification (return 5 top IDs person predicted).

1. For the one model: P1, P2, P3, P4, P5.
2. For the multi-models: P1, P2, P3, P4, Top5.

3.2.3.1 Matching of OMS

In the matching step, through the result of prediction, that is taken from the One model. The result of prediction is taken for the greatest probabilities which appear in the form of a list of a text containing specific information, e.g. a class of this model is named by 'L1' that contains the information, Footwear trace, for an adult person, includes more foot, and Straight type feet direction. To produce one final list as a list of arguments for the matching function, we extract the list of information from the prediction result.

The matching task is a search for matching each element of this final list with the list of information that already Known (figure 3.2). After calculating the percentage of congruence (score), we conclude the final result of the system that is determined by the score. the final result help system on recognizing if the input image of HFT was known by the system, i.e. it Known in the system because the models trained on it, or not, i.e. it does not exist in the system and it is for an unknown person for the system but it identifies the similar information of it. Algorithm 1 present matching algorithm for OMS.

Algorithm 1 Matching in OMS

```
Data:
P1,                                          ◁Partofpredictionfortheonemodel
P2,                                          ◁Partofpredictionfortheonemodel
P3,                                          ◁Partofpredictionfortheonemodel
P4,                                          ◁Partofpredictionfortheonemodel
P5,                                          ◁Probabilityoftheonemodelprediction
Result:
result                                       ◁Infoofthesuitableperson(ID, Score, Probability)
Initialization:
 ID ← 0
 score ← 0
 prob ← 0
 list ← EmptyListOf2D
 dataInfo ← EmptyListOf2D                    ◁Informationpersonsfromfilesystem
 dataInfo← readFile("PATH_FILE_SYSTEM")
 prob ← P5
 for i ← 1, dataInfo.Length do
    ID ← i;
     if dataInfo[i][1]=P1 then
        score ← score+25;
    end
    if dataInfo[i][2]=P2 then
        score ← score+25;
    end
    if P3="MoreFoot" then
        if dataInfo[i][3]=P4 then
            score ← score+25;
        end
    end
    score ← score+25;
    list[i] ← ID;
    list[i][i + 1] ← score;
    list[i][i + 2] ← prob;
end
OrderList( list, 2);                         ◁Orderlistbycolunm2
result← list[0];
```

3.2.3.2 Matching of MMS

In the matching step, through the list of prediction results that are taken from the Barefoot/Footwear model, and Age group model, and One/More foot model, or Feet direction model, the prediction results are taken for the greatest probabilities. In addition to the list of 5 top prediction results that are taken from the ID classification model. To produce one final list as a list of arguments for the matching function, we concatenate the list of prediction results with the list of 5 top prediction results. The final list consists of the five top outputs of the ID classification with the outputs of the Barefoot/Footwear model, Age group model, and Feet direction model.

The matching task is a search for matching each element of this predicted list with the list of information that already Known, (figure 3.2). After calculating the percentage of congruence (score), we conclude the final result of the system that is determined by the score and the greatest probability of the ID person. the final result help system on recognizing if the input image of HFT was known by the system, i.e. it Known in the system that can identify this person because the models trained on it, or not, i.e. it does not exist in the system and it is for an unknown person for the system but it identifies the similar person with information on it. This function returns one result that is the choosed person among the 5 candidates persons with his information and score. Algorithm 2 present matching algorithm for MMS.

Algorithm 2 Matching in MMS

```
Data:
P1,                    ◁Thepredictedlabelofmodel1(Barefoot/Footwear)
P1_prob,               ◁Predictionprobabilityofmodel1(Barefoot/Footwear)
P2,                    ◁Thepredictedlabelofmodel2(Agegroup)
P2_prob,               ◁Predictionprobabilityofmodel2(Agegroup)
P3,                    ◁Thepredictedlabelofmodel3(One/Morefoot)
P3_prob,               ◁Predictionprobabilityofmodel3(One/Morefoot)
P4,                    ◁Thepredictedlabelofmodel4(Feetdirection)
P4_prob,               ◁Predictionprobabilityofmodel4(Feetdirection)
Top5,                  ◁Top5predictionslist
Result:
result                 ◁Infoofthesuitableperson(ID, Score, Probability)
Initialization:
 ID ← 0
 score ← 0
 ID_prob ← 0           ◁PredictionprobabilityofIDclassificationmodel
 listTop5← EmptyListof2D
 dataInfo ← EmptyListof2D          ◁Informationpersonsfromfilesystem
 dataInfo← readFile("PATH_FILE_SYSTEM")
 for i ← 1, 5 do
    ID ← Top5[i];
     prob ← Top5[i][1];
     if dataInfo[ID]=P1 then
        score ← score + P1_prob*25;
    end
    if dataInfo[ID]=P2 then
        score ← score + P2_prob*25;
    end
    if P3="MoreFoot" then
        if dataInfo[ID]=P4 then
            score ← score+P4_prob *25;
        end
    else
        score ← score+P3_prob *25;
    end
    listTop5[i] ← ID;
    listTop5[i][i + 1] ← score;
    listTop5[i][i + 2] ← ID_prob;
end
OrderList( listTop5, 2);          ◁OrderlistTop5bycolunm2
result← listTop5[0];
```

3.2.4 Making a decision

After the matching function is finished, we get the person most expected with the information and their score of matching. So, we can decide from this score and probability of the prediction model, if the input image is for a known person or not, i.e. Known in the system or not.

$res \leftarrow$ **Matching_Function**($P1$,$P2$,$P3$,$P4$,$Var5$);
$ID \leftarrow res[1]$;
$score \leftarrow res[2]$;
$probability \leftarrow res[3]$;
if $score > 50$ **AND** $probability > 0.5$ **then**
  return $Known$
else
  return $Unknown$
end if

Conclusion

In this chapter, we proposed two systems architectures for recognizing HFT. The first system is one model based system, while the second is a multi-models based system. Firstly, we collected and processed our dataset, then built datasets for models to be trained in our proposed systems. Also, we explained the most important steps for building and training models. In the next chapter, we will discuss the results of implementing these proposed HFT systems.

Chapter 4

Results and discussions

Introduction

After we have seen dataset preparation, systems architectures and also a model comparative experiment. In this chapter, we indicate the important tools chosen for the development of the systems. Besides, we will present in details the results of the tests. Finally, we discuss the results of testing, to decide which system is better than the other system in terms of accuracy value. We concluded our study by showing the interfaces of developed applications, respectively, for web application and mobile application.

4.1 Implementation tools

In this section, we present the important development tools that we used in our work. These tools are used in two phases, training phase and deployment phases. First, we talk about the development tools used in brief, then we explain each tool separately by explaining the programming languages used, their working environment, API Framework, and the most important libraries applied in addition to the operating system used in the meantime.

- ✓ **Phase 1**: training models, the principle language of programming used is Python3 in the Environment Colaboratory by using the TensorFlow2.x library to train and test the models. Besides, use the Drive cloud to upload the dataset from it (Mount Drive) to the colab environment for training.
- ✓ **Phase 2**: deployment, loading models and using it as a development application, we developed two applications, web application and mobile application.
    1. Development of web application:

    - Front-end: HTML5, CSS3, JS3.
    - Back-end: Python 3.6.0.
    - Library: TensorFlow 2.3.0.
    - API Framework: Flask 1.1.1.
    - Environment: PyCharm 2020.2.1.

2. Development of mobile application:
   - Front-end: Dart 2.9.0.
   - Back-end: Dart 2.9.0.
   - Library: TFLite 2.3.0.
   - API Framework: Flutter 1.21.
   - Environment: Android studio 4.0.1.

Programming with Python is more important as compared to the operating system (Windows, Lunix, Mac). Among the OS, Linux is one of the most popular platforms suitable for programming. Linux makes Python easy because it doesn't go through many installation steps, unlike Windows. Because Python comes pre-installed in Linux distributions, Ubuntu, Debian, ...etc. Besides, it is easy to switch between versions of Python when working in Linux, and many more reasons. Therefore, we see Linux as a better choice than other OS. We are working on Linux with two different distributions, Ubuntu and Debian on different laptops. Where the web application was programmed in the first laptop (Ubuntu), and the mobile application was programmed in the second laptop (Debian).

1. Laptop1:
   - Ubuntu 18.04.4 LTS.
   - Processor Intel® Core™ i7-5500U CPU @ 2.40GHz × 4.
   - Graphics Intel® HD Graphics 5500 (Broadwell GT2).
   - GNOME 3.28.2.
   - OS type 64-bit.
   - Disk 1.2 TB.
2. Laptop2:
   - Debian GNU/Linux 10 (buster).
   - Processor Intel® Core™ i5-6200U CPU @ 2.30GHz × 4.
   - Graphics Intel® HD Graphics 520 (Skylake GT2).
   - GNOME 3.30.2.
   - OS type 64-bit.
   - Disk 209.5 GB.

Now, we explain important development tools that are mentioned previously, which are Python, Colab, TensorFlow, Flask, HTML, CSS, JS, Dart, Flutter, TFLite.

- **Python** is high-level language programming that was created by Guido van Rossum. It is open-source, free, fast, and easy-to-use. Python is extensively used in the artificial intelligence and data science field which is developed in machine learning and deep learning projects. This programming language provides excellent library support in especially AI science such as [88, 58]:

- Python libraries for AI: AIMA, PyDatalog, SimpleAI, EasyAi.

- Python libraries for ML: Keras, NumPy, Scikit-learn, PyBrain, MDP, PyML.

- Python libraries for DL: TensorFlow, Spark-deep-learning, CNTK, Caffe.

- **TensorFlow** is an open-source library and a Python-friendly free for numerical computation and large-scale, Created by the Google Brain team that makes DL and ML faster and easier. TensorFlow can train and run deep neural networks for image recognition, recurrent neural networks, handwritten digit classification, natural language processing, word embeddings, sequence-to-sequence models for machine translation. . . ,and more [89].

- **Colaboratory** (colab or CO) is a Google research project. Colab is a free and a hosted Jupyter notebook environment that requires no setup to use and runs entirely in the cloud (by the browser), it was created to help disseminate machine learning education and research [90, 91]. Colab hardware available:
  - Accelerator GPU: up to Tesla K80 with 12 GB of GDDR5 VRAM, Intel Xeon Processor with 2cores @2.20 GHz, RAM 13 GB .
  - Accelerator TPU: Cloud TPU with 180 teraflops of computation, Intel Xeon Processor with 2cores @2.30 GHz, RAM 13 GB.
  - Processor: CPUs Intel Xeon Processor with 2cores @2.30 GHz, RAM 13 GB.
  - Disk: 68,40 GB free space.
  - RAM: 12,72 GB free space.

- **Drive**: is one of the cloud storage services developed by Google, it allows users to store files on their servers which get 15 GB of space in Drive for free. its main purpose is to expand the ability to store files beyond the limits of the local hard drive [92].
- **Flask**: is one of the frameworks for Python, this means flask provides us with tools, libraries, and technologies that allow us to build a web application. Flask is also called a micro web framework because it does not require particular tools or libraries [93].
- **HTML**: is the HyperText Markup Language which is used for creating attractive web pages and web applications. HTML is used to structure a web document and to create a front-end web application [94].
- **CSS**: is stands for Cascading Style Sheets used for describing the presentation of a document HTML (how HTML elements are to be displayed).
- **JS**: (Javascript) is a scripting or programming language that enables us to create dynamically updating content and implement complex features on web pages [94].

- **Dart**: is programming language has recently incorporated a set of extensions designed to support asynchrony and generator functions[95]. It is an open-source general-purpose programming language that was developed by google.
- **Flutter**: is a new cross platform mobile application SDK offered by Google which lets developers build beautiful Android and iOS applications using a common codebase.
- **TFLite**: is a cross-platform library that helps to deploy TFLite models onto mobile devices. TensorFlowLite interpreter for running inference on a model.

4.2 Implementation results

In this part, we implement the experiment on all models for each system, which pass by two phases, a training phase, and a testing phase. The training phase to train the architecture model on a specific dataset, and the testing phase to know the accuracy of the trained model. After starting the implementation, we define some configuration arguments with constant values in all models:

- Training data: 80%.
- Validation data: 20%.
- Testing data: 20%.
- INIT_LR: 1e-3.
- K_neighbors: 3.
- Batch size (BS): 10.
- Optimizer: Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)

These parameters will apply in all models, with the split of the dataset (training, testing, validation) that will be the same splitting. Table 4.1 below presents the values of splitting each dataset to 3 sets, training, testing, validation.

Dataset	**Dataset size 100%**	**Training set 80%**	**Testing set / Validation set 20%**
One model	673	538	135
Barefoot/Footwear	673	538	135
Barefoot Age group	216	173	43
Footwear Age group	457	366	91
Barefoot One/More foot	216	173	43
Footwear One/More foot	457	366	91
Barefoot Feet direction	164	131	33
Footwear Feet direction	308	246	62
Barefoot ID classification	216	173	43
Footwear ID classification	457	366	91

Table 4.1: Splitting dataset to training set, testing set and validation set.

The following diagrams 4.1, 4.2 represents the splitting dataset in 3 sets, training set (80%), testing set (20%), validation set (20%).

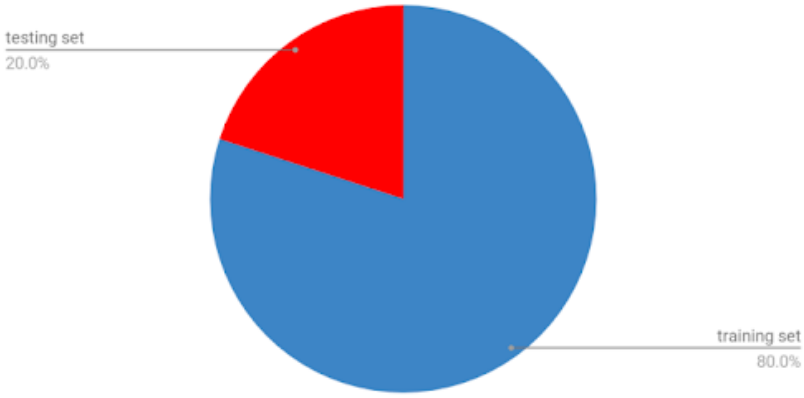


Figure 4.1: Split training set and testing set of each dataset.

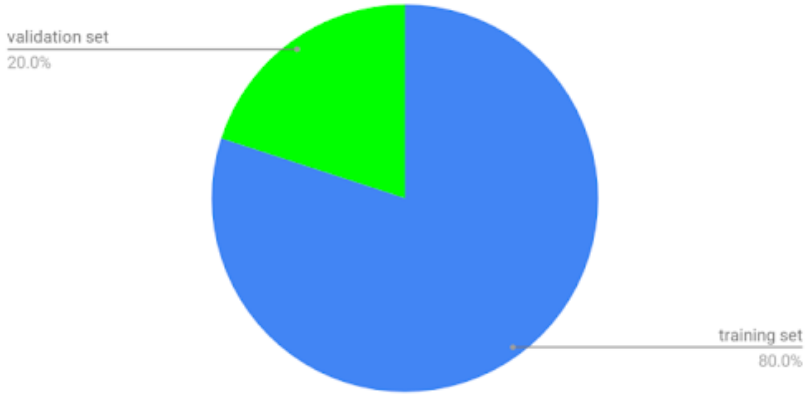


Figure 4.2: Split training set and validation set of each dataset.

From the above diagrams, we have 3 values, the value of the training set ratio (80%), the value of the testing (20%), and the value of the validation set ratio (20%). Intentionally, we set the same values for the ratios of all dataset to confirm the validity of the comparison because we will be doing a comparative study of two types of systems that based on different models, so this comparative study does not consider the changing in the ratio as a factor to determine the best model (the best system). For this, we chose the best ratio value for the best model and apply it across all models. We chose these values after trying different values on the models.

4.2.1 Training phase

The training phase is the most important phase, which trains the proposed model architecture, Xception (features extractor CNNs) with classifier (Softmax, SVM, KNN). We mention that the experiment performs for 3 cases of classifier (Softmax, SVM, KNN), and the training phase is for classifier only (features extractor CNNs untrainable). After training the classifier of each model, we get 10 trained models (one for OMS and 9 for MMS) for each case. In our work, we proposed 2 different systems so we training the

model(s) of each system on its training dataset, as follows:

- **For one model system**:

  - the proposed One model architecture trained by One model dataset.

- **For one model system**:
  - the proposed Barefoot/Footwear architecture trained by the Barefoot/Footwear dataset.
  - the proposed Barefoot Age group architecture trained by the Barefoot Age group dataset.
  - the proposed Footwear Age group architecture trained by the Footwear Age group dataset.
  - the proposed Barefoot One/More foot architecture trained by the Barefoot One/More foot dataset.
  - the proposed Footwear One/More foot architecture trained by the Footwear One/More foot dataset.
  - the proposed Barefoot Feet direction architecture trained by the Barefoot Feet direction dataset.
  - the proposed Footwear Feet direction architecture trained by the Footwear Feet direction dataset.
  - the proposed Barefoot ID classification architecture trained by the Barefoot ID classification dataset.
  - the proposed Footwear ID classification architecture trained by the Footwear ID classification dataset.

4.2.2 Testing phase

In the previous phase, we get a set of trained models. For examining the validity of each model, we test these models on its testing dataset. The factor used in comparison is accuracy as an evaluation metric. The results of the testing phase are shown in Table 4.2, for each system in 3 cases of classifier (Softmax, SVM, KNN).

Name model	Number of classes	Epoch	Accuracy for Softmax Classifier	Accuracy for SVM Classifier	Accuracy for KNN Classifier
One model	24	100	71.00	76.40	67.97
Barefoot / Footwear	2	70	97,22	96.52	88.88
Barefoot Age group	3	150	88.00	82.00	84.00
Footwear Age group	3	100	88.42	83.15	82.10
Barefoot One / More foot	2	150	92,00	86.00	74.00
Footwear One / More foot	2	70	91,75	95.87	87.62
Barefoot Feet direction	3	150	94,00	75.75	72.72
Footwear Feet direction	3	100	87,10	79.03	79.03
Barefoot ID classification	22	150	80.60	79.45	76.71
Footwear ID classification	38	150	95,06	89.50	72.22

Table 4.2: Results of testing the trained model (s) OMS & MMS.

From the previous table, we notice that the OMS get the best accuracy value with the SVM classifier, while get the worst accuracy value with the KNN classifier, and the Softmax classifier present medium value of accuracy.

In the MMS, we create a chart line that presents the 3 curves of the cases for the MMS according to the previous results of table 4.2. The green curve is the results of the assigned value for the softmax classifier, the blue curve is the results of the assigned value for the SVM classifier, and the red curve is the results of the assigned value for the KNN classifier, as shows chart line 4.3.

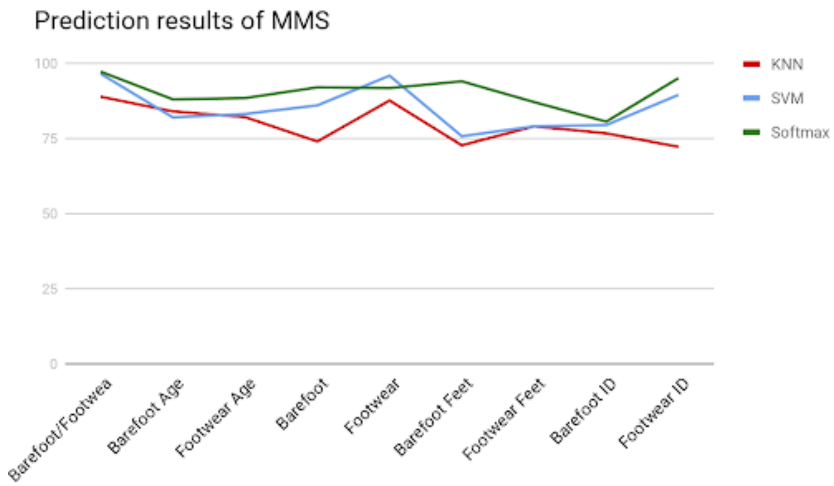


Figure 4.3: Prediction results of different classifiers for MMS.

From the ChartLine 4.3, we distinguish three curves of three classifications:

- The curve of Softmax classifier has better accuracy values then the others classifier, with the exception of accuracy value for the Footwear One/More model, so that the SVM classifier gets a slightly greater accuracy value than it.
- SVM classifier has better accuracy values then the KNN classifier only, and in the case of accuracy value for the Footwear One/More model is better then the Softmax classifier.

- KNN classifier has worse accuracy values then the others classifier, and in the case of accuracy value for the Footwear One/More model is better then the SVM classifier. Table 4.3 illustrates the summarized results of the previous results of table 4.2. The table below shows the final accuracy for each system in 3 cases of classifier (Softmax, SVM, KNN).

System Type	Accuracy for Softmax Classifier	Accuracy for SVM Classifier	Accuracy for KNN Classifier
One Model System	71.00%	76.40%	67.97%
Multi-model System	90.46%	85.25%	79.69%

Table 4.3: Summarized results of testing OMS & MMS.

Through the table above , we can see so clearly the important results that help us to compare so easily and to decide the best system. Where:

- In the one models system:
  - The accuracy for the softmax classifier is 71.00%.
  - The accuracy for the SVM classifier is 76.40%.
  - The accuracy for the KNN classifier is 67.97%.

In the MMS, the accuracy of the models was taken as the average accuracy of each of them. Where this accuracy was calculated through the following equation: Where, i: model number, i$\in \{1, ..., 9\}$, $N : totalnumberofmodels, N = 9$

- In the multi-model system:
  - The accuracy for the softmax classifier is 90.46%.
  - The accuracy for the SVM classifier is 85.25%.
  - The accuracy for the KNN classifier is 79.69%.

4.3 Results discussion

Based on the previous results and the analyses of these results, in the multi-models system, we get **90.46%** accuracy value with the Softmax classifier that is better than the others classifiers.While in the one model system, we get **76.40%** accuracy value with the SVM classifier that is better than the other classifiers. So, when comparing the OMS with MMS, we conclude the MMS accuracy values with all classifiers are better than the OMS accuracy values for all classifiers.

We can conclude the decision of the comparative study. For the "Human foot trace recognition system", we consider the second experiment the best, e.i. The multi-models system is more successful and reliable than the one model system for the Softmax classifier.

4.4 Application interface

In this section, we set our work in an integrated and directly usable application. First, we save and load the selected models in the application (web, mobile). Then, we developed our applications suitable for these models. There are two sides for each application, OMS, OMM as shown in figures 4.4, 4.5.

Figure 4.4: Home interface for web app.

Figure 4.5: Home interface for mobile app.

4.4.1 One model system interface

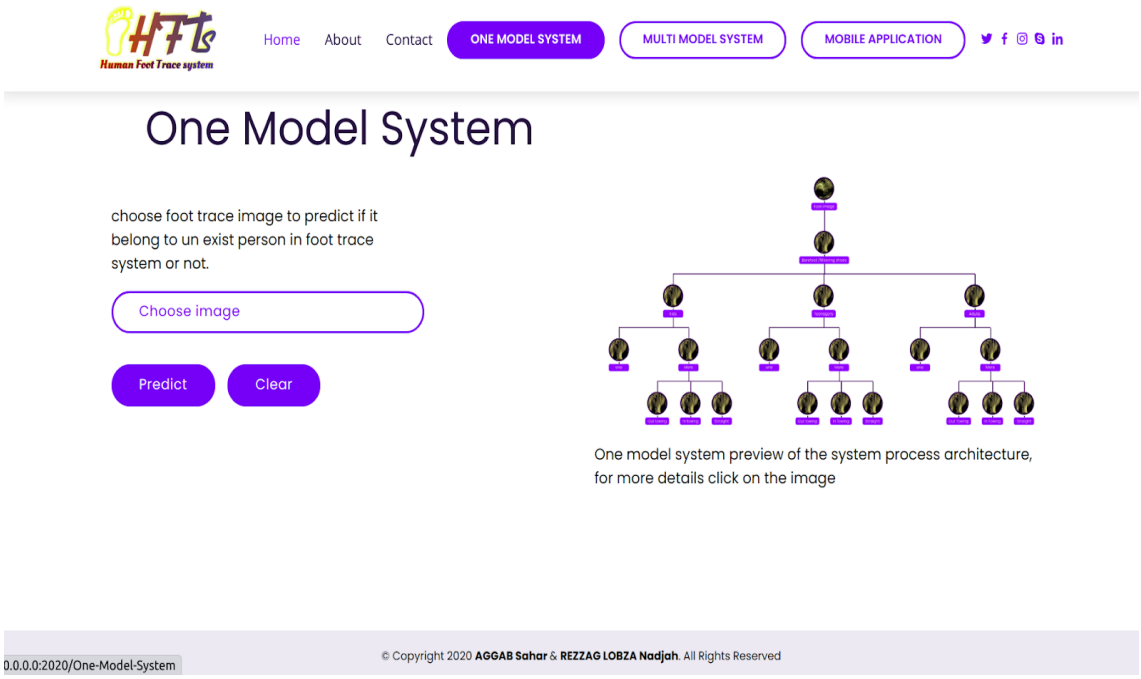


Figure 4.6: OMS home interface for web app.

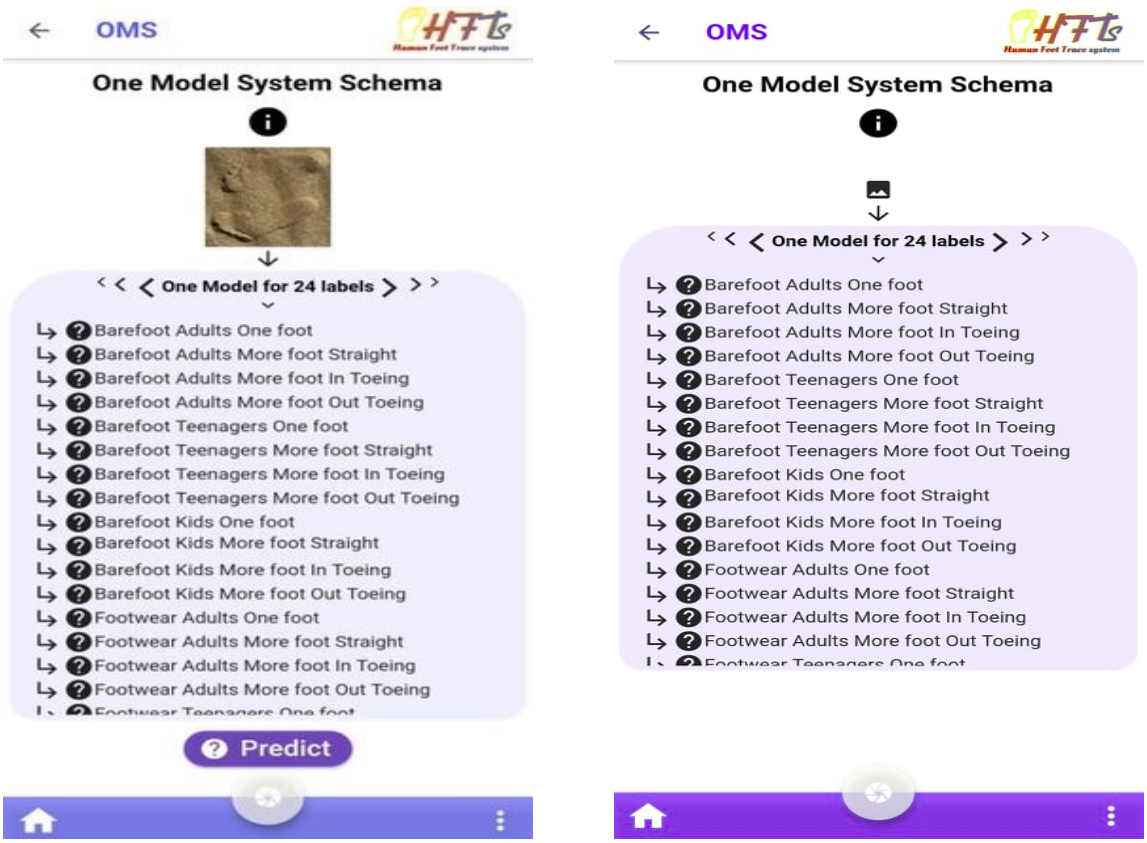


Figure 4.7: OMS home interface for mobile app.

Figures 4.6, 4.7 illustrate the important components of these OMS home interfaces applications, which are:

- **Button for load image**: allow to upload an image by drag and drop or click for web app, and by picker from gallery or taken image through camera.
- **Button for predicted image**: start prediction by the model, and show the picker image with the final predicted results of HFT in table, as shown in figures 4.8, 4.9.

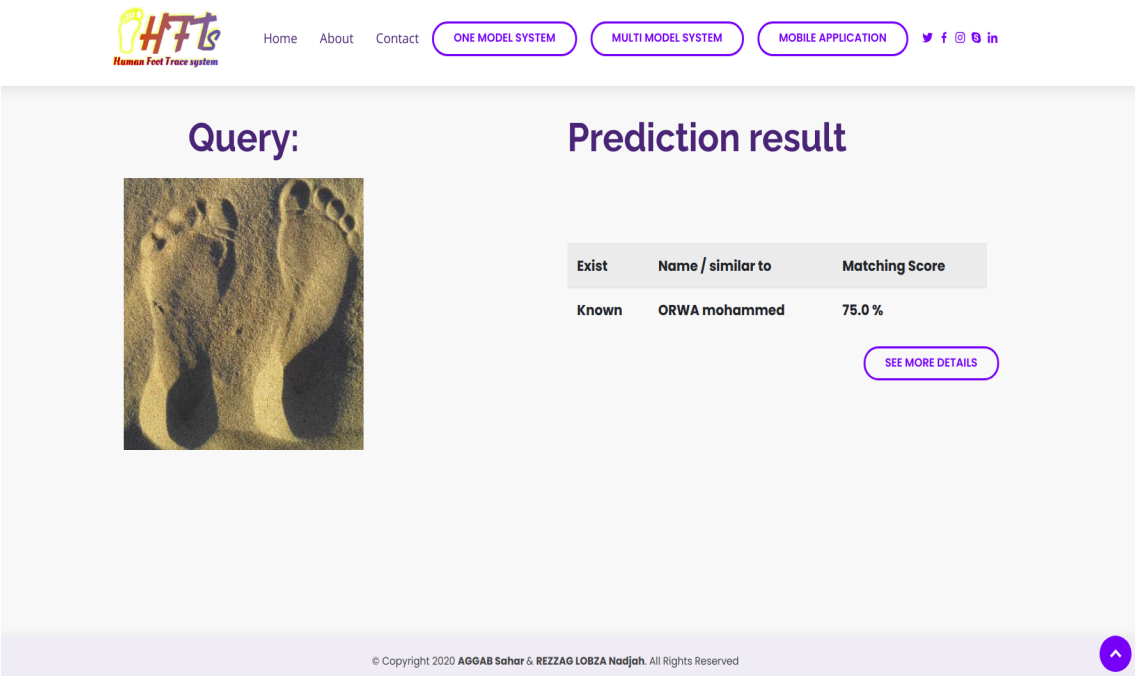


Figure 4.8: OMS prediction results for the web app.

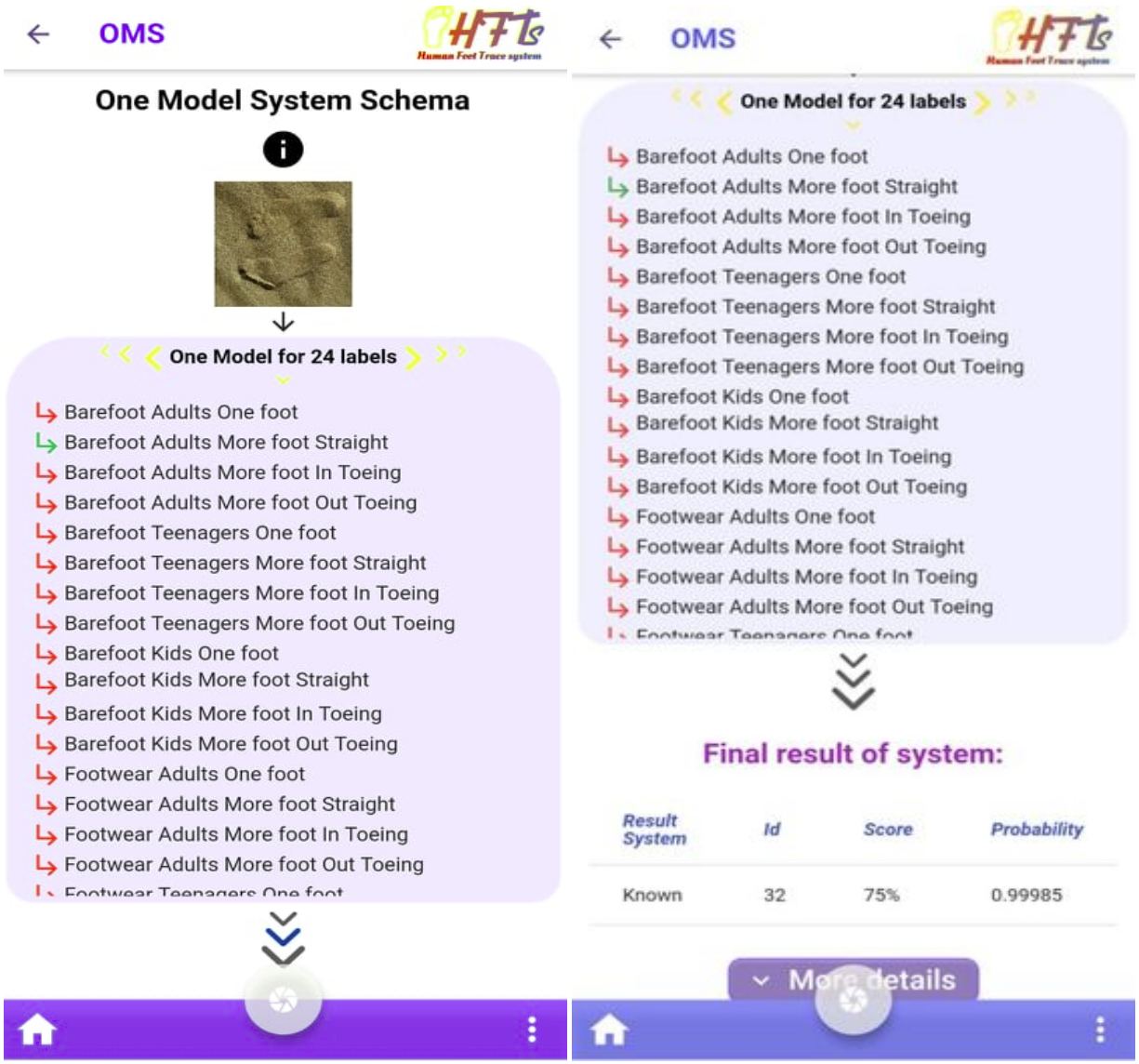


Figure 4.9: OMS prediction results for the mobile app.

Figures 4.8, 4.9 illustrate another component (new button) that is see more details. Button for more details: detailed the predictions results of HFT in a set of tables, as shown in the figures 4.10, 4.11.

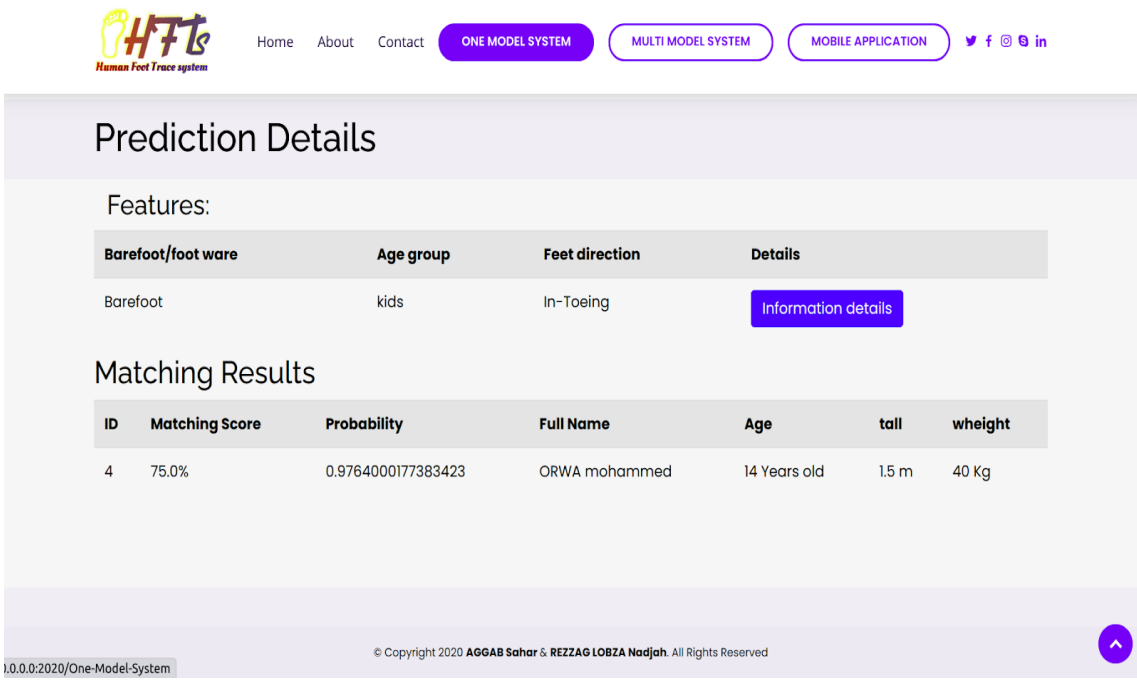


Figure 4.10: OMS details prediction results for the web app.

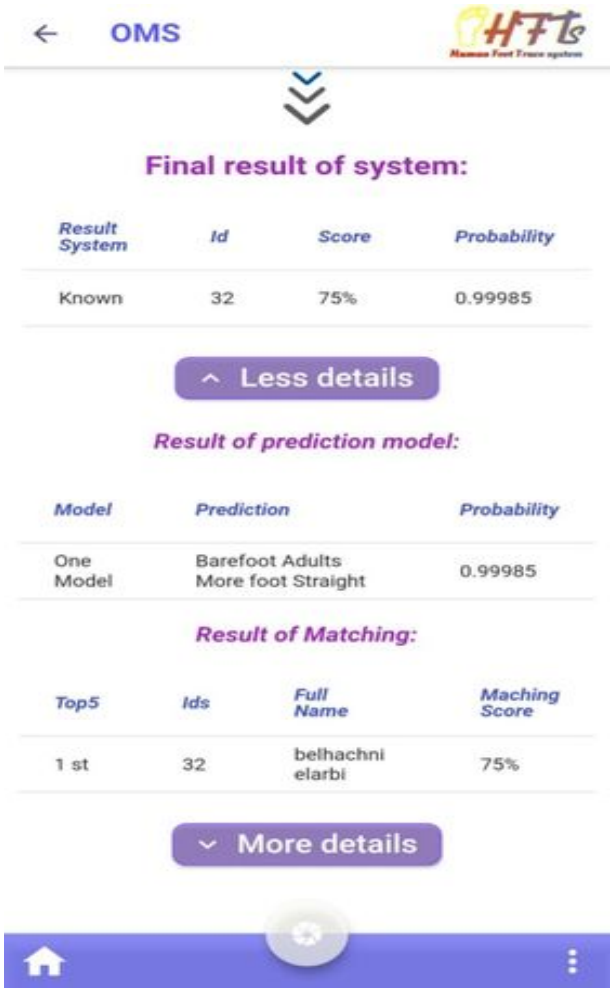


Figure 4.11: OMS details prediction results for the mobile app.

- **Table for features extraction**: from prediction result, extracting the important features, Barefoot/Footwear, Age group, Feet direction.
- **Table for matching result**: is the final output result after getting the most suitable person. Besides, show the detailed information (ID, full name, Age, tall, weight) of the suitable person by clicking on the button Information details, as shown in the figures 4.12, 4.13.

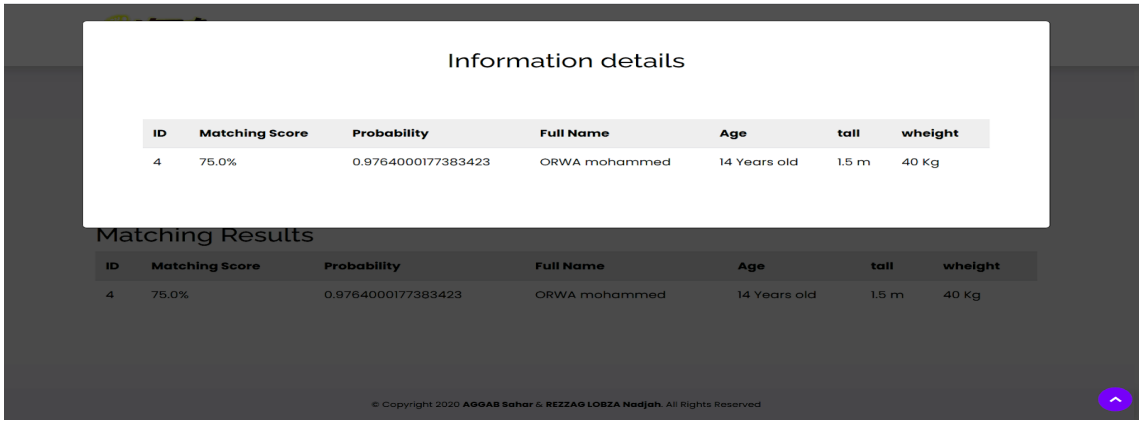


Figure 4.12: OMS information of suitable person for the web app.

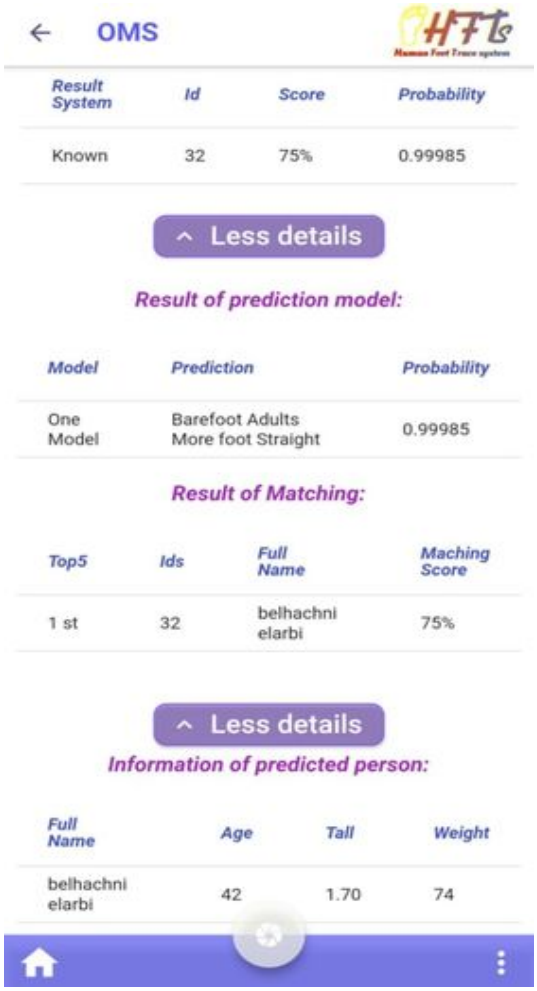


Figure 4.13: OMS information of suitable person for the mobile app.

4.4.2 Multi-models system interface

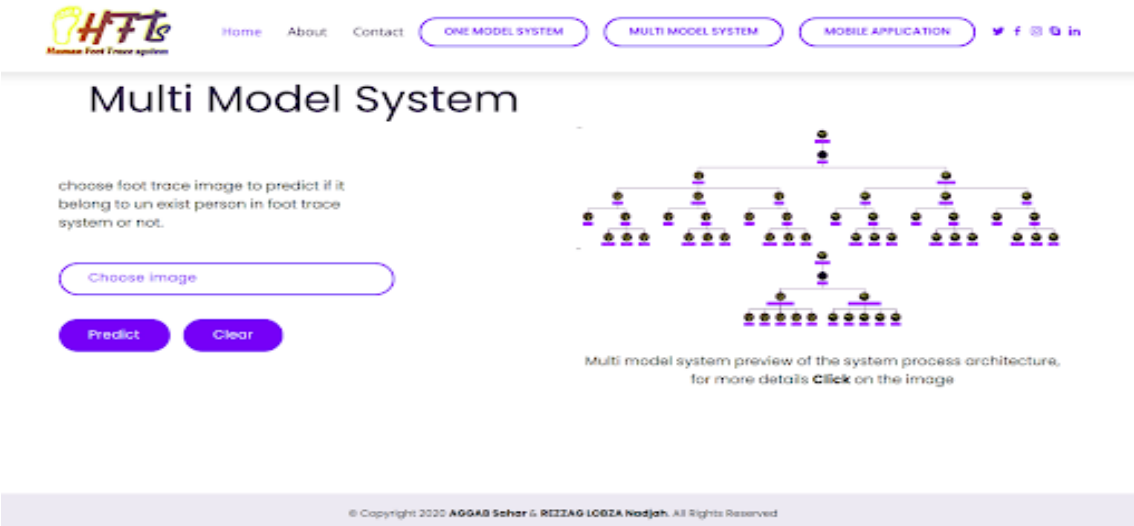


Figure 4.14: MMS home interface for web app.

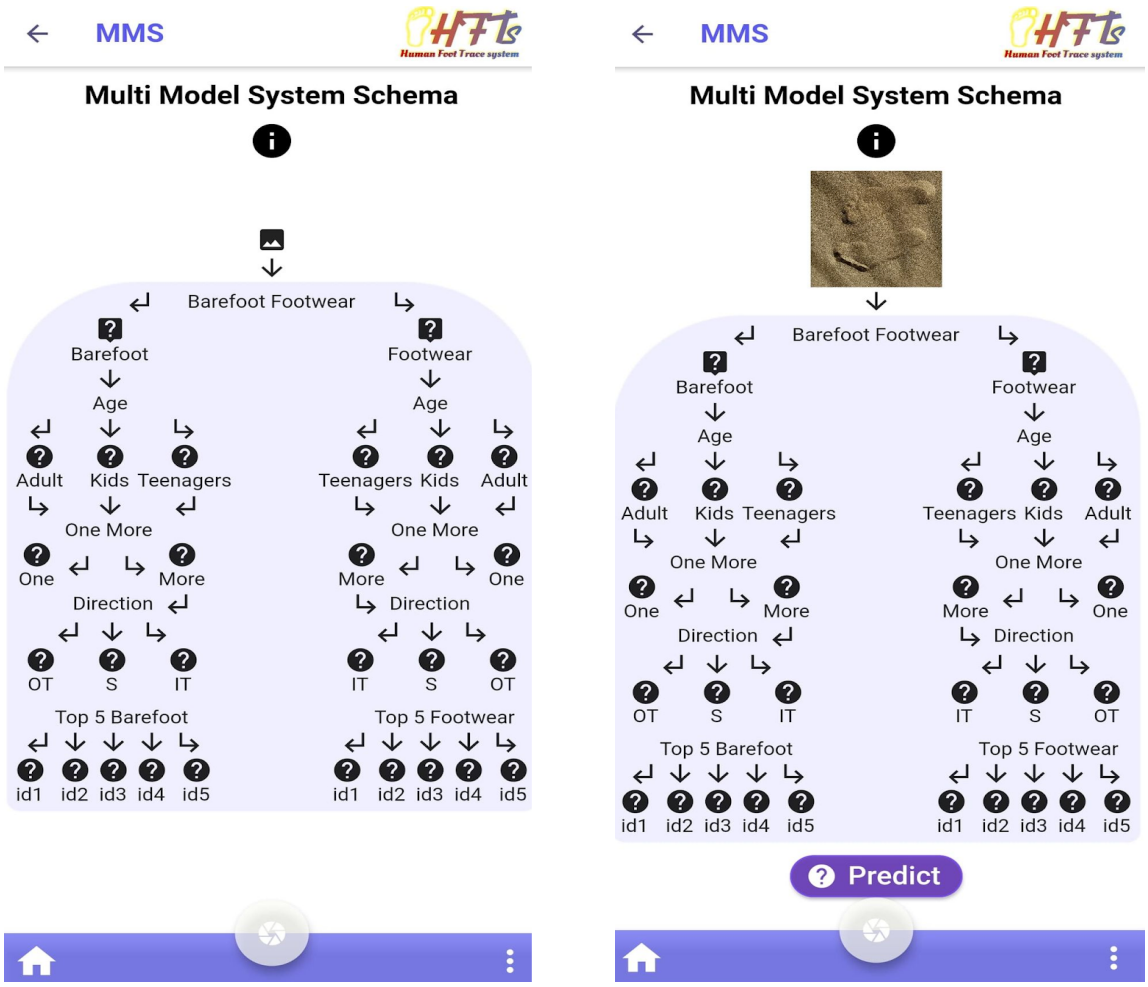


Figure 4.15: MMS home interface for mobile app.

Figures 4.14, 4.15 illustrate the important components of these MMS home interfaces applications, which are:

- Button for load image: allow to upload an image by drag and drop or click for web app, and by picker from gallery or taken image through camera.
- Button for predicted image: start prediction by the model, and show the picker image with the final predicted results of HFT in table, as shown in figures 4.16, 4.17.

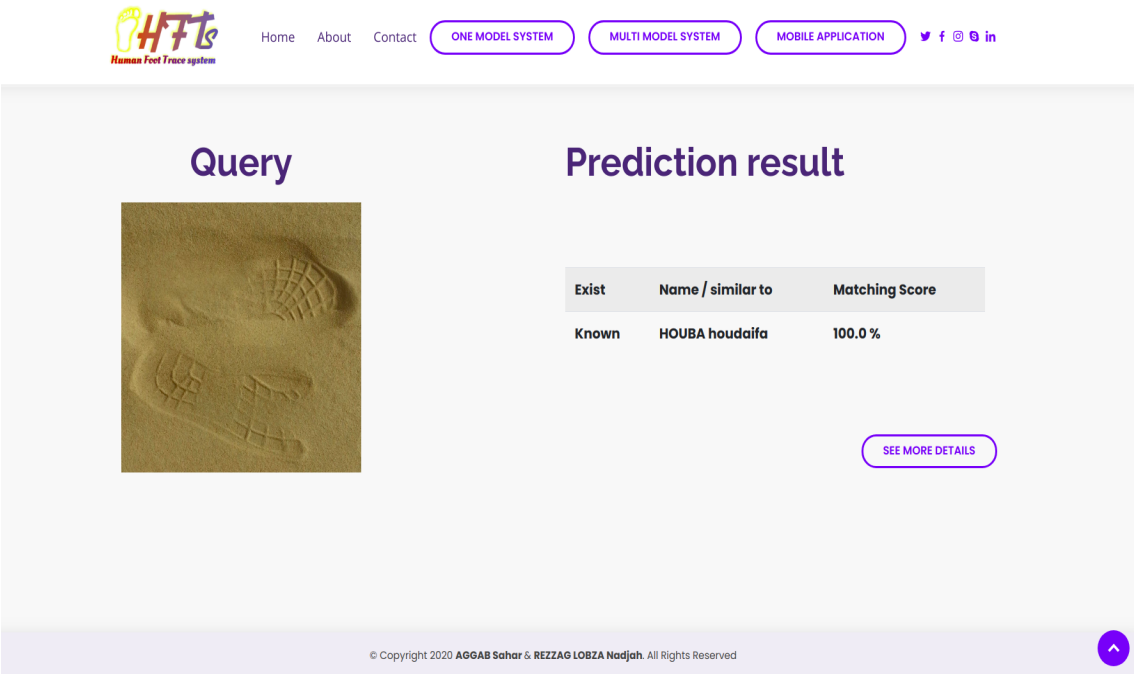


Figure 4.16: MMS prediction results for the web app.

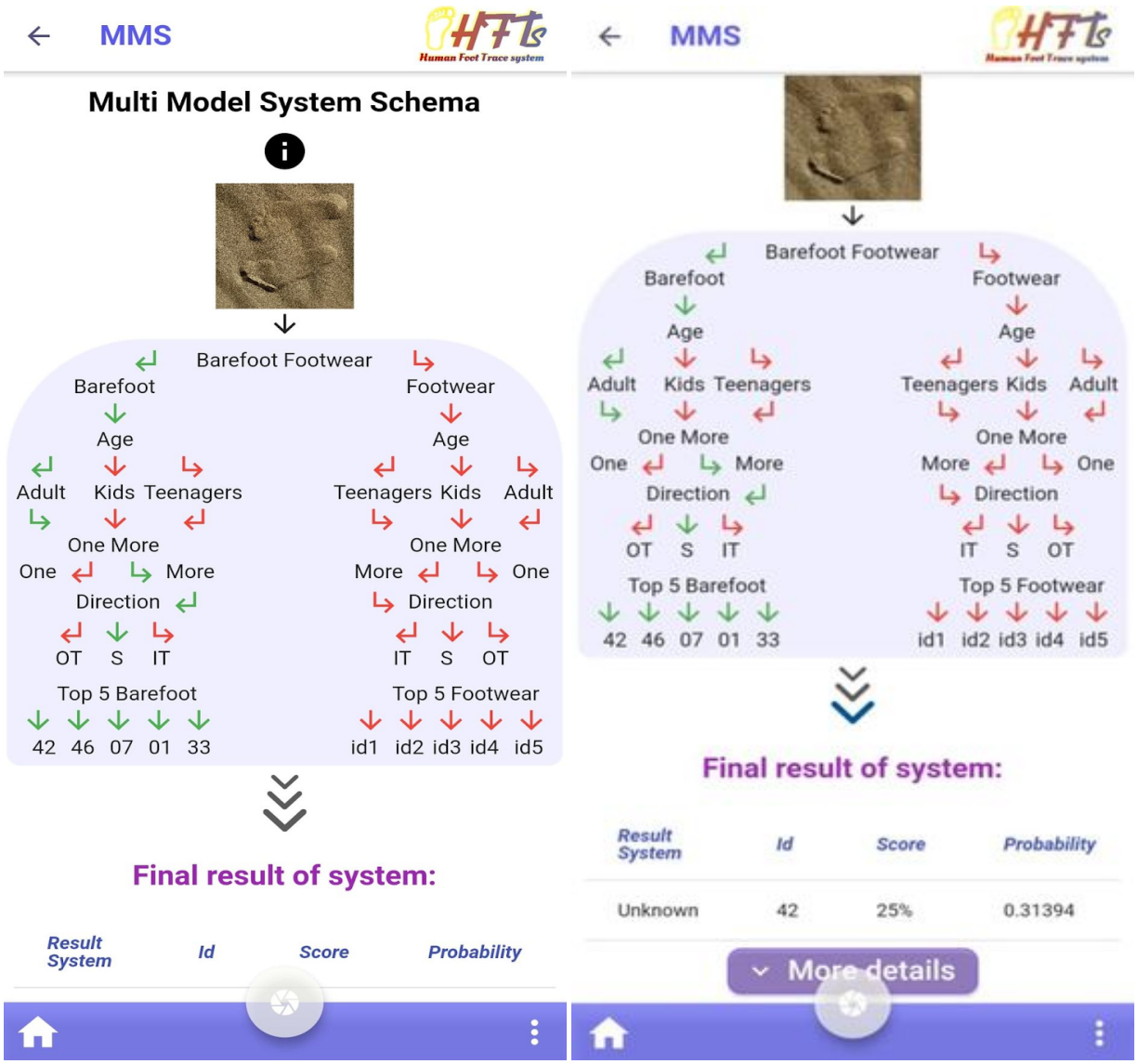


Figure 4.17: MMS prediction results for the mobile app.

Figures 4.16, 4.17 illustrate another component (new button) that is see more details.

- Button for more details: detailed the predictions results of HFT in a set of tables, as shown in the figures 4.18, 4.19.

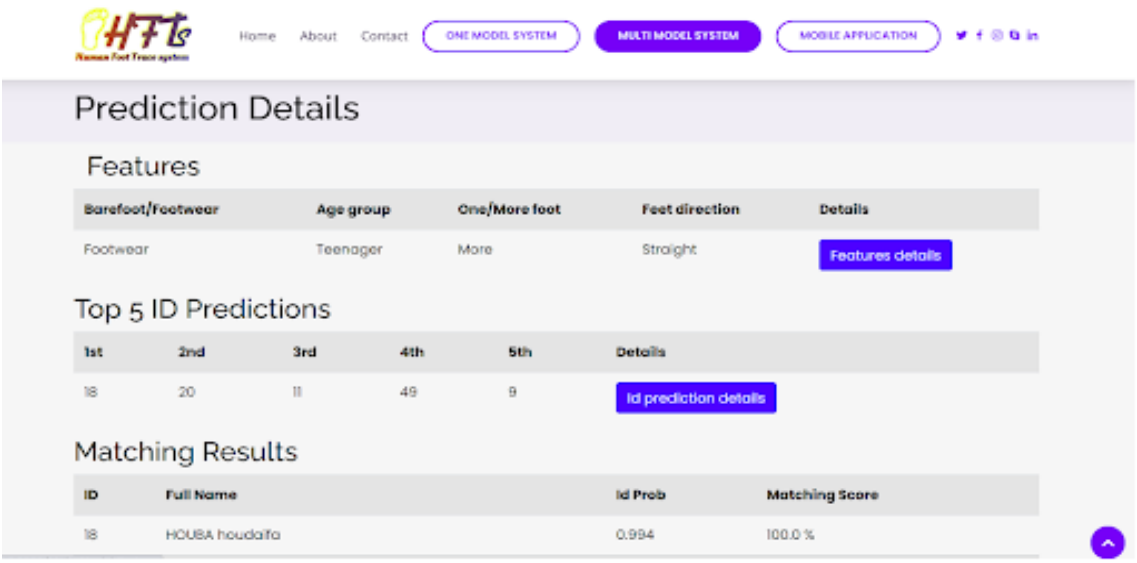


Figure 4.18: MMS details prediction results for the web app.

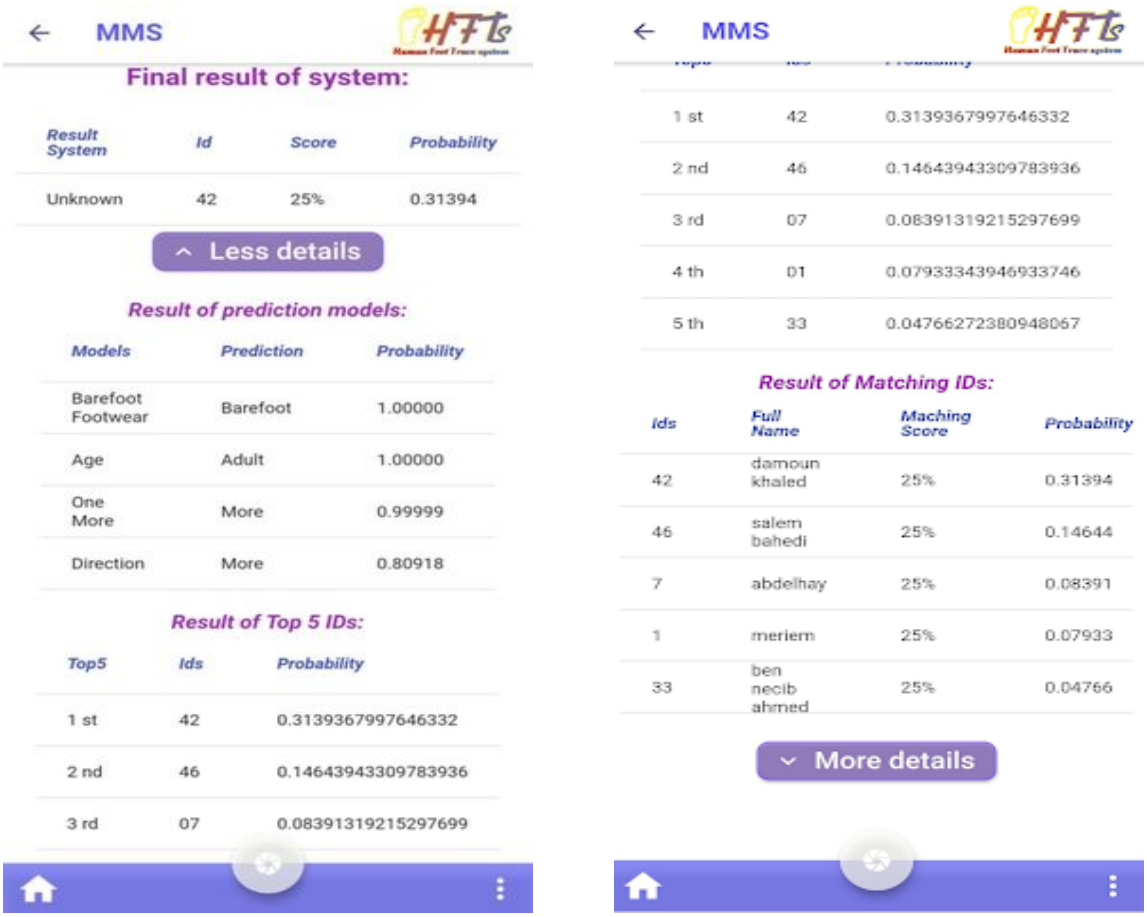


Figure 4.19: MMS details prediction results for the mobile app.

- **Table for features extraction**: from predictions results, getting the important features, Barefoot/Footwear, Age group, Feet direction.
- **Table for 5 top id similar person**: from prediction the model ID classification, getting 5 top id classification from 5 the best probabilities. Besides, show these 5 probabilities by clicking on the button ID prediction details, as shown in the figures 4.20, 4.21.

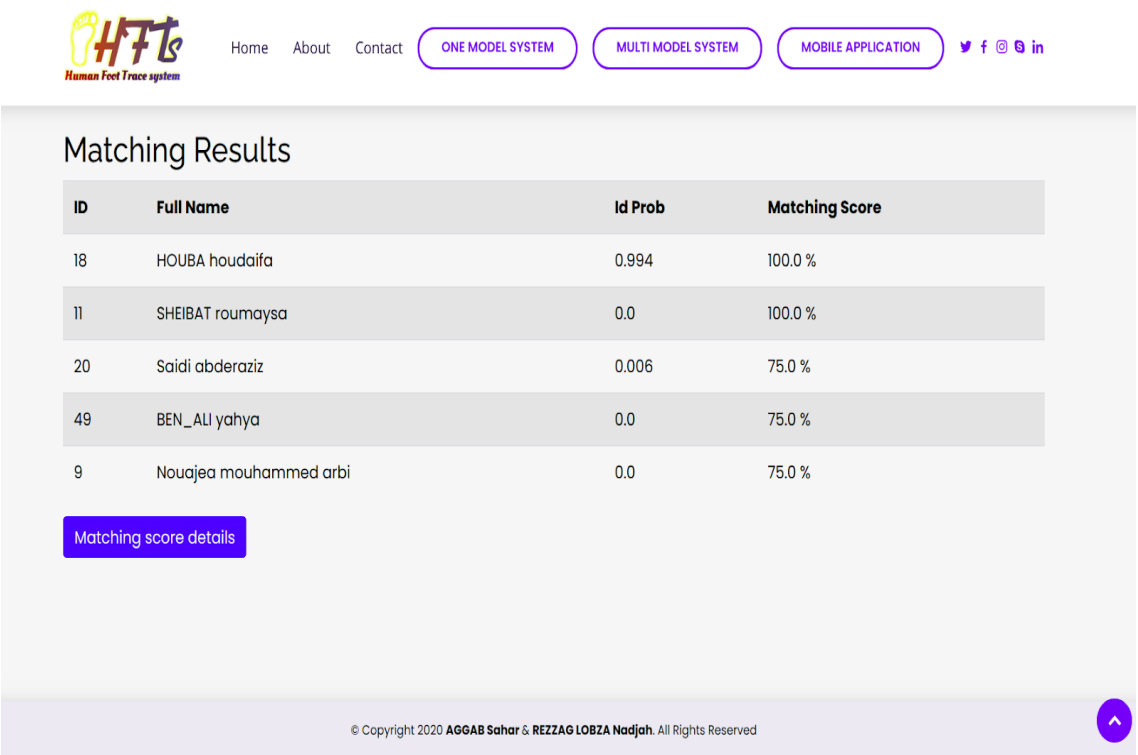


Figure 4.20: MMS details prediction of 5 top persons for the web app.

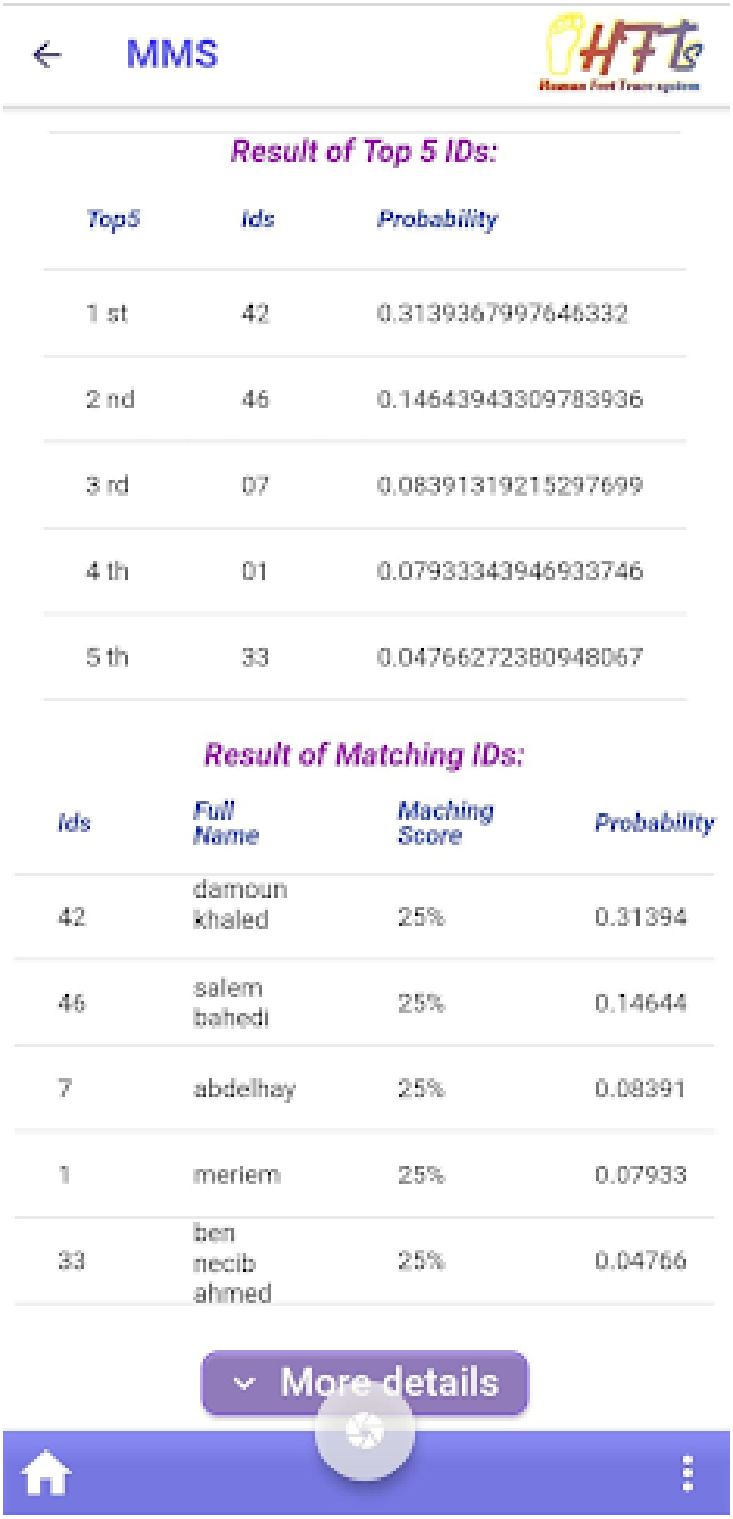
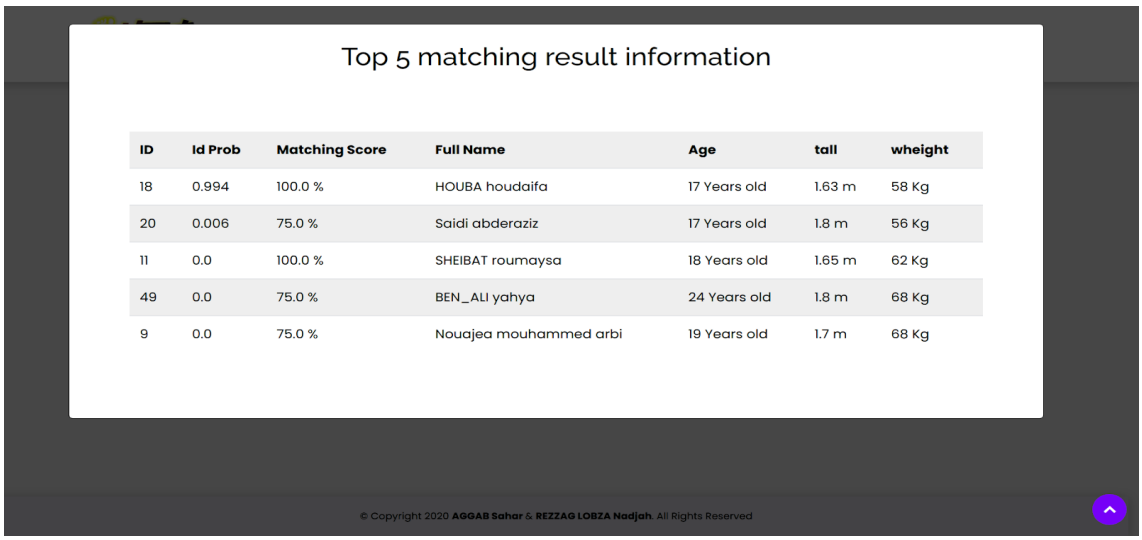


Figure 4.21: MMS details prediction of 5 top persons for the mobile app.

- **Table for matching result**: is the 5 outputs results after getting the 5 id similar person with scores of matching. Besides, show the detailed information (ID, full name, Age, tall, weight) of each person and the detailed process of matching by clicking on the button Matching score details, as shown in the figures 4.22, 4.23.

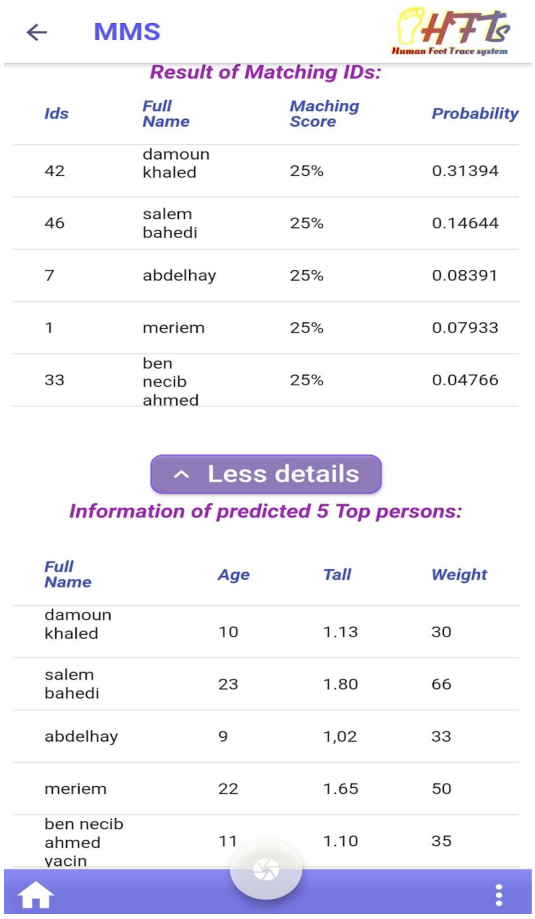


Top 5 matching result information

ID	Id Prob	Matching Score	Full Name	Age	tall	wheight
18	0.994	100.0 %	HOUBA houdaifa	17 Years old	1.63 m	58 Kg
20	0.006	75.0 %	Saidi abderaziz	17 Years old	1.8 m	56 Kg
11	0.0	100.0 %	SHEIBAT roumaysa	18 Years old	1.65 m	62 Kg
49	0.0	75.0 %	BEN_ALI yahya	24 Years old	1.8 m	68 Kg
9	0.0	75.0 %	Nouajea mouhammed arbi	19 Years old	1.7 m	68 Kg

© Copyright 2020 AGGAB Sahar & REZZAG LOBZA Nadjah. All Rights Reserved

Figure 4.22: MMS details of matching score information for the web app.



MMS

Result of Matching IDs:

Ids	Full Name	Maching Score	Probability
42	damoun khaled	25%	0.31394
46	salem bahedi	25%	0.14644
7	abdelhay	25%	0.08391
1	meriem	25%	0.07933
33	ben necib ahmed	25%	0.04766

Less details

Information of predicted 5 Top persons:

Full Name	Age	Tall	Weight
damoun khaled	10	1.13	30
salem bahedi	23	1.80	66
abdelhay	9	1,02	33
meriem	22	1.65	50
ben necib ahmed yacin	11	1.10	35

Figure 4.23: MMS details of matching score information for the mobile app.

Conclusion

In this chapter, we presented the development tools for training and testing the proposed systems for human foot trace recognition. From the experiments of out comparative study, we concluded that the best system after selection is the one with softmax. Finally, we conclude that the multi-models system is more reliable than the One model system.

Conclusion and future work

The trace recognition system is one of the most important systems which help in searching for missing people by taking pictures remotely from hard-to-reach places, using means such as a Drone camera [21]. This system relied on advanced techniques such as CNNs to process and recognize images in the computer vision field.

In this work, we proposed a system that recognizes images of the human foot traces in the desert. For that, we conducted a comparative study of two different systems architectures, the first one is based on one model approach, and the other one is based on a multi-models approach for 3 types of classifiers softmax, SVM, and KNN. As a result, the system based on multi-models approach is the best efficient and accurate with the softmax classifier, and it achieved an accuracy of **90.46%**.

As a new system in the study, we faced many difficulties and challenges. For better performance and a better pace of work, it would be good to take into consideration these difficulties. Those challenges can be categorized as follows:

- Dealing with dataset images: Preparing images took the largest part of work time. When splitting datasets into several parts, the number of images in the last partial datasets reduces, which leads to a decrease in the accuracy of the trained models in this partial dataset. Therefore, the volume of data is small which is insufficient to train a model efficiently [19].
- Dealing with many models: the number of models in the system is important because it affects the overall size of the system. If the number of models is very large, it consumes more space for storage and processing. Therefore, an in-depth study of its structure and its data size must be conducted to improve system performance during the collaborative work of these models, instead of taking feet traces pictures each person, can record a video of every foot trace, and then extract a large set of the images from this video, for collecting the largest possible number of images in the shortest time. In the future, there is a possibility of improving this system based on improving its models by getting more data and making the data balance in each class for any dataset, or by modifying its architecture.

Bibliographie

[1] "Artec 3d scanners help police identify footprints at crime scenes," https://www.artec3d.com/cases/identifying-footprints-forensics-21st-century, 2020, [Online; accessed 04-03-2020].

[2] F. Cheng, "Detection of melanotic cooked shrimp using computer vision," *International Agricultural Engineering Journal*, **25**(2016), pages 276–285.

[3] M. Elgendy, *Deep Learning for Vision Systems*, Manning Publications2020, ISBN 9781617296192, URL https://books.google.dz/books?id=6gkLzAEACAAJ.

[4] G. siva, "Machine learning and its algorithms," https://medium.com/@gayathrisiva8608/machine-learning-and-its-algorithms-b58b4bcfaaab, May 11, 2019, [Online; accessed 22-05-2020].

[5] M. A. T. Jimenez, "Summarization of video from feature extraction method using image processing & artificial intelligence," .

[6] Y. Tang, "Enhancements of major aerosol components due to additional hono sources in the north china plain and implications for visibility and haze," *Advances in Atmospheric Sciences*, **30**(2013), no. 1, pages 57–66.

[7] N. Altman, "Machine learning: supervised methods," , 2018.

[8] "Semi-supervised," http://primo.ai/index.php?title=Semi-Supervised, 8 December 2019, [Online; accessed 03-09-2020].

[9] D. Matolak, "A machine learning approach for power allocation in hetnets considering qos," "2018 IEEE International Conference on Communications (ICC)," , IEEE2018, pages 1–7.

[10] M. Tripathy, "Low snr speech enhancement with dnn based phase estimation," *International Journal of Speech Technology*, **22**(2019), no. 1, pages 283–292.

[11] F. Faria, "Deep feature-based classifiers for fruit fly identification (diptera: Tephritidae)," 2018, pages 41–47, doi:10.1109/SIBGRAPI.2018.00012.

[12] T. Kanchan, "Bare footprint metric analysis methods for comparison and identification in forensic examinations: A review of literature," *Journal of Forensic and Legal Medicine*, **58**(2018), pages 101–112.

[13] T. J. U. Thompson, *Forensic Human Identification: An Introduction*, Taylor & Francis2007.

[14] S. Castillo Pinel, "Forensic podiatry: A new path for podiatrists?" (2015).

[15] A. Baliyan, *Applications of Deep Learning and Big IoT on Personalized Healthcare Services*, Advances in Medical Technologies and Clinical Practice, IGI Global2020, ISBN 9781799821021, URL https://books.google.dz/books?id=u8jYDwAAQBAJ.

[16] A. C. Berg, "Low-power image recognition challenge," "2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)," , IEEE2017, pages 99–104.

[17] Y. Bengio, *Learning deep architectures for AI*, Now Publishers Inc2009.

[18] M. Ramakrishnan, "Multiple feature extraction for foot print image," *National Journal of System and Information Technology*, **5**(2012), no. 2, page 142.

[19] A. Zaib, "Deep learning for medical image processing: Overview, challenges and the future," "Classification in BioApps," , Springer2018, pages 323–350.

[20] E. M. Pagliaro, "Forensic evidence and crime scene investigation," *Journal of Forensic Investigation*, **1**(2013), no. 2, pages 1–5.

[21] D. L. Newman, "Drone for collecting images and system for categorizing image data," , October 23 2014, uS Patent App. 13/998,871.

[22] G. Liu, *Meshfree Methods: Moving Beyond the Finite Element Method, Second Edition*, Series in Cellular and Clinical Imaging, CRC Press/Taylor & Francis Group2009, ISBN 9781439853832, URL https://books.google.dz/books?id=wa2NAgAAQBAJ.

[23] C. E. Ferguson, *Forensic criminology*, Academic Press2009.

[24] B. F. Goings, "Illuminating footwear assembly," , November 30 2017, uS Patent App. 15/357,704.

[25] W. H. Marg, "Dictionary. com," *Keyword: Persistence*, (2011).

[26] M. Staff, *Merriam-Webster's Collegiate Dictionary: Eleventh Edition*, Logos Bible Software, Merriam-Webster, Incorporated2004, ISBN 9780877798095, URL https://books.google.dz/books?id=TAnheeIPcAEC.

[27] G. Chauvy, *Les archives de la police scientifique française: des origines à nos jours*, Presses de la Cité2013, ISBN 9782258106109, URL https://books.google.fr/books?id=cjL8ngEACAAJ.

[28] O. Eldjabari, "Al-firasah kawasilah lilaistidlal ealaa al-shakhsiah fi mujtamae wadi souf -tatabie al'athar wa al-qiafat 'anamudhijina- [physiognomy as a means of inferring personality in the community of wadi suf trace impact and hospitality two models]," (2014).

[29] M. Malik, *English Translation of the Meaning of Al-Qur'an: The Guidance for Mankind (English Only)*, Institute of Islamic Knowledge1997, ISBN 9780911119770, URL https://books.google.dz/books?id=PqhECQAAQBAJ.

[30] C. Sharma, "Impact assessment of textile industries of rajasthan using geo-spatial techniques," "Impact of Textile Dyes on Public Health and the Environment," , IGI Global2020, pages 70–89.

[31] Z. Mohamed, "Islam between human inspiration and divine revelation: A counter-argument to the theory of cultural vacuum by hourani," *PRICE: RM 35.00*, (2011), page 173.

[32] H. Haggrén, "Real-time photogrammetry as used for machine vision applications," *The Canadian Surveyor*, **41**(1987), no. 2, pages 201–208.

[33] P. Meer, "Robust techniques for computer vision," *Emerging topics in computer vision*, (2004), pages 107–190.

[34] V. Loia, *Advances in Computer Science and Ubiquitous Computing: CSA-CUTE2016*, volume 421, Springer2016.

[35] J. C. Rodríguez-Quiñonez, *Developing and Applying Optoelectronics in Machine Vision*, IGI Global2016.

[36] A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*, " O'Reilly Media, Inc."2008.

[37] S. Gollapudi, "Deep learning for computer vision," "Learn Computer Vision Using OpenCV," , Springer2019, pages 51–69.

[38] L. Xiaobo, *Computer Vision: Systems, Theory And Applications: Selected Papers From Vision Interface 1992*, volume 38, World Scientific1993.

[39] T. Kasbe, *Information Technology & MIS*, Idea Publishing2018, URL https://books.google.dz/books?id=e6xwDwAAQBAJ.

[40] R. Boyle, *Image processing, analysis, and machine vision*, Cengage Learning2014.

[41] M. Priya, "Study of image recognition using cellular associated artificial neural networks," "World Congress on Engineering 2012. July 4-6, 2012. London, UK.", volume 2188, International Association of Engineers2010, pages 514–518.

[42] P. A. Laplante, *Dictionary of computer science, engineering and technology*, CRC Press2000.

[43] Z. Zain, *Proceedings of the 11th National Technical Seminar on Unmanned System Technology 2019: NUSYS'19*, SPRINGER VERLAG, SINGAPOR2020, ISBN 9789811552816, URL https://books.google.dz/books?id=t_LvDwAAQBAJ.

[44] D. Ryan, *E-learning modules: DLR Associates Series*, AuthorHouse2012.

[45] A. Kumar, *Challenges and Applications for Implementing Machine Learning in Computer Vision*, Advances in Computer and Electrical Engineering, IGI Global2019, ISBN 9781799801849, URL https://books.google.dz/books?id=RPO9DwAAQBAJ.

[46] S. Kapoor, *Intelligent Computing: Proceedings of the 2019 Computing Conference, Volume 2*, Advances in Intelligent Systems and Computing, Springer International Publishing2019, ISBN 9783030228682, URL https://books.google.dz/books?id=2m2hDwAAQBAJ.

[47] D. Shanmugapriya, *Handbook of Research on Machine and Deep Learning Applications for Cyber Security*, Advances in Information Security, Privacy, and Ethics, IGI Global2019, ISBN 9781522596134, URL https://books.google.dz/books?id=vK2rDwAAQBAJ.

[48] E. Alpaydin, *Introduction to Machine Learning*, Adaptive Computation and Machine Learning series, MIT Press2020, ISBN 9780262043793, URL https://books.google.dz/books?id=tZnSDwAAQBAJ.

[49] P. Suresh, "Multi label text classification and the effect of label pyramid clustering," .

[50] et al., *Consumer-Driven Technologies in Healthcare: Breakthroughs in Research and Practice*, IGI Global2018.

[51] V. V. Estrela, *Deep learning for image processing applications*, volume 31, IOS Press2017.

[52] J. Faytong, "Machine learning and event-based software testing: Classifiers for identifying infeasible gui event sequences," "Advances in Computers," volume 86, Elsevier2012, pages 109–135.

[53] A. Shmilovici, "Support vector machines," "Data mining and knowledge discovery handbook," , Springer2009, pages 231–247.

[54] S. K. Prabhakar, *KNN Classifier and K-Means Clustering for Robust Classification of Epilepsy from EEG Signals. A Detailed Analysis*, diplom. de2017.

[55] et al., *Information Computing And Automation (In 3 Volumes)-Proceedings Of The International Conference*, World Scientific2008.

[56] C.-F. Tsai, "The distance function effect on k-nearest neighbor classification for medical datasets," *SpringerPlus*, **5**(2016), no. 1, pages 1–9.

[57] B. Wikipedians, *Computer Science*, PediaPress, URL https://books.google.dz/books?id=Yte2cXVES9EC.

[58] L. Hluchỳ, "Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey," *Artificial Intelligence Review*, **52**(2019), no. 1, pages 77–124.

[59] C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," "Advances in Neural Information Processing Systems," 2019, pages 5049–5059.

[60] Y. H. Liu, "Feature extraction and image recognition with convolutional neural networks," "Journal of Physics: Conference Series," volume 1087, 2018, page 062032.

[61] U. Kose, "Artificial intelligence and applied mathematics in engineering problems," .

[62] V. Estrela, *Deep Learning for Image Processing Applications*, Advances in Parallel Computing, IOS Press2017, ISBN 9781614998228, URL https://books.google.dz/books?id=vsFVDwAAQBAJ.

[63] E. Corchado, *International Joint Conference SOCO'18-CISIS'18-ICEUTE'18: San Sebastián, Spain, June 6-8, 2018 Proceedings*, Advances in Intelligent Systems and Computing, Springer International Publishing2018, ISBN 9783319941202, URL https://books.google.dz/books?id=I_ZeDwAAQBAJ.

[64] F. Ortuño, *Bioinformatics and Biomedical Engineering: 8th International Work-Conference, IWBBIO 2020, Granada, Spain, May 6-8, 2020, Proceedings*, LNCS sublibrary: Bioinformatics, Springer2020, ISBN 9783030453855, URL https://books.google.dz/books?id=nIvgDwAAQBAJ.

[65] Y. Chen, *Intelligent System Design: Proceedings of Intelligent System Design: INDIA 2019*, Advances in Intelligent Systems and Computing Series, Springer Singapore Pte. Limited2020, ISBN 9789811554001, URL https://books.google.dz/books?id=XL_2DwAAQBAJ.

[66] B. Durakovic, "Deep transfer learning for human identification based on footprint: A comparative study," *Periodicals of Engineering and Natural Sciences*, **7**(2019), no. 3, pages 1300–1307.

[67] P. Marcelino, "Transfer learning from pre-trained models," *Towards Data Science*, (2018).

[68] Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, **22**(2009), no. 10, pages 1345–1359.

[69] S. Chinnamgari, *R Machine Learning Projects: Implement supervised, unsupervised, and reinforcement learning techniques using R 3.5*, Packt Publishing2019, ISBN 9781789806090, URL https://books.google.dz/books?id=4dKDDwAAQBAJ.

[70] A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, (2014).

[71] M. U. Hassan, "Vgg16–convolutional network for classification and detection," , 2018.

[72] A. Rosebrock, "Imagenet: Vggnet, resnet, inception, and xception with keras," *Mars*, (2017).

[73] J. Sun, "Deep residual learning for image recognition," "Proceedings of the IEEE conference on computer vision and pattern recognition," 2016, pages 770–778.

[74] A. Rabinovich, "Going deeper with convolutions," "Proceedings of the IEEE conference on computer vision and pattern recognition," 2015, pages 1–9.

[75] B. Kwolek, "Deep cnn-based recognition of jsl finger spelling," "International Conference on Hybrid Artificial Intelligence Systems," , Springer2019, pages 602–613.

[76] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," "Proceedings of the IEEE conference on computer vision and pattern recognition," 2017, pages 1251–1258.

[77] P. Nicolas, *Scala for Machine Learning*, Community experience distilled, Packt Publishing2015, ISBN 9781783558759, URL https://books.google.dz/books?id=d5EIBgAAQBAJ.

[78] H. Farhadi, *Machine Learning: Advanced Techniques and Emerging Applications*, IntechOpen2018, ISBN 9781789237528, URL https://books.google.dz/books?id=EG-QDwAAQBAJ.

[79] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," (2011).

[80] J. G. Burrow, "The use of the podotrack in forensic podiatry for collection and analysis of bare footprints using the reel method of measurement," *Science & Justice*, **56**(2016), no. 3, pages 216–222.

[81] T. Kanchan, "Two-dimensional metric comparison between dynamic bare and sock-clad footprints for its forensic implications–a pilot study," *Science & Justice*, **59**(2019), no. 1, pages 46–51.

[82] T. Kanchan, "A metric study of insole foot impressions in footwear of identical twins," *Journal of Forensic and Legal Medicine*, **52**(2017), pages 116–121.

[83] S. Jones, "The identification and reliability of static and dynamic barefoot impression measurements: A systematic review," *Forensic science international*, **289**(2018), pages 156–164.

[84] J. A. DiMaggio, *Forensic podiatry: principles and methods*, CRC Press2017.

[85] L. Trabut, *Étude sur l'halfa, Stipa Tenacissima*, A. Jourdan1889.

[86] A. Hasan, "Dynamic weights equations for converting grayscale image to rgb image," *Journal of University of Babylon for Pure and Applied Sciences*, **26**(2018), doi:10.29196/jubpas.v26i8.1677.

[87] R. C. Thomas, *Computer vision: A first course*, Blackwell Scientific Publications, Ltd.1988.

[88] D. Kuhlman, *A Python Book: Beginning Python, Advanced Python, and Python Exercises*, PLATYPUS GLOBAL MEDIA2011, ISBN 9780984221233, URL https://books.google.dz/books?id=1FL-ygAACAAJ.

[89] et al., *Query-Focused Abstractive Summarization using Neural Networks*, Ph.D. thesis, Lethbridge, Alta.: University of Lethbridge, Dept. of Mathematics and . . . , 2019.

[90] E. Bisong, *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, Springer2019.

[91] L. Massaron, *Deep Learning For Dummies*, Wiley2019, ISBN 9781119543046, URL https://books.google.dz/books?id=JjqRDwAAQBAJ.

[92] S. Bhowmik, *Cloud Computing*, Cambridge University Press2017, ISBN 9781316638101, URL https://books.google.dz/books?id=jeTFDgAAQBAJ.

[93] G. Dwyer, *Flask By Example*, Community experience distilled, Packt Publishing2016, ISBN 9781785283482, URL https://books.google.dz/books?id=M7PjCwAAQBAJ.

[94] R. Huddleston, *HTML, XHTML, and CSS: Your visual blueprint for designing effective Web pages*, Visual Blueprint, Wiley2009, ISBN 9780470396117, URL https://books.google.dz/books?id=gjtOPP_v7J0C.

[95] G. Bracha, "Spicing up dart with side effects," *Queue*, **13**(2015), no. 3, pages 40–59.