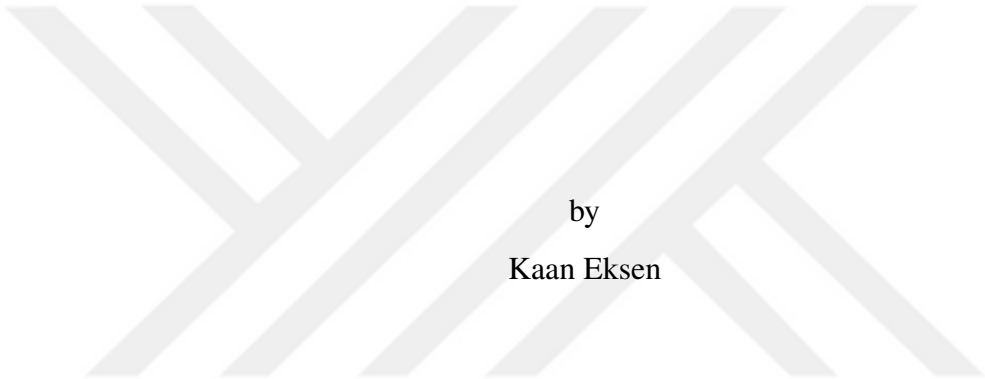


NOVEL MOBILE PRE-DIAGNOSTIC SOLUTION FOR PES PLANUS AND PES
CAVUS USING IMAGE PROCESSING AND DEEP NEURAL NETWORKS



by
Kaan Eksen

Submitted to Graduate School of Natural and Applied Sciences
in Partial Fulfillment of the Requirements
for the Degree of Master of Science in
Computer Engineering

Yeditepe University
2022

NOVEL MOBILE PRE-DIAGNOSTIC SOLUTION FOR PES PLANUS AND PES
CAVUS USING IMAGE PROCESSING AND DEEP NEURAL NETWORKS

APPROVED BY:

Assist. Prof. Dr. Tacha Serif
(Thesis Supervisor)
(Yeditepe University)

.....

Assist. Prof. Dr. Onur Demir
(Yeditepe University)

.....

Prof. Dr. Fatih Uğurdağ
(Özyeğin University)

.....

DATE OF APPROVAL:/....../20....

I hereby declare that this thesis is my own work and that all information in this thesis has been obtained and presented in accordance with academic rules and ethical conduct. I have fully cited and referenced all material and results as required by these rules and conduct, and this thesis study does not contain any plagiarism. If any material used in the thesis requires copyright, the necessary permissions have been obtained. No material from this thesis has been used for the award of another degree.

I accept all kinds of legal liability that may arise in case contrary to these situations.

Name, Last Name

Kaan Eksen

Signature

.....

ACKNOWLEDGEMENTS

I would like to express my most profound gratefulness to Yeditepe University, Computer Engineering faculty member Dr. Tacha Serif. I would also like to thank Dr. Safa Serif for his help during the tests and his suggestions on my thesis.

I would also thank my dear family, who have always stood by and supported me in every way.

I would like to thank my dear wife, Melis Eldem Eksen, who always motivates, supports me and has never-ending love.



ABSTRACT

NOVEL MOBILE PRE-DIAGNOSTIC SOLUTION FOR PES PLANUS AND PES CAVUS USING IMAGE PROCESSING AND DEEP NEURAL NETWORKS

According to previous research in this domain, about thirty percent of the world population has some kind of foot deformity. Among all the deformities, pes planus produced by a lack of the medial longitudinal arch of the foot and pes cavus produced by an excessively high plantar longitudinal arch are the foot deformities that most adversely affect the community's quality of life. Bearing in mind the above, the proposed solution aims to pre-diagnose pes planus and pes cavus through a mobile phone application using image processing and deep neural networks with the help of traditional deformity recognition methods existing in the literature. To this end, this work is composed of three studies that build upon each other, to improve the foot deformity detection precision. Accordingly, the first prototype is implemented using machine learning techniques to estimate pes planus and pes cavus using footprint approach and tested 34 participants. To benchmark this prototype, an orthopedist is asked to provide key decision-making points manually so that a comparative calculation can be performed. The comparison results of the two showed that the expert and prototype findings were in agreement with each other more than 90 percent. As a follow up, another deformity classification method is implemented as a second study, which entailed, rearfoot angle and input from the healthcare professionals. This prototype was tested with 9 participants. Within the scope of the second study, data entries were provided by using the prototype by healthcare professionals. The system results were compared with the deformities detected by the healthcare professionals, and it was observed that the results were matched at a rate of 27.7. In the final study, the detection method developed in the first prototype is improved and tested with the test set used in the second study. The results showed that deformities detected by the physician and the system matched 83.3 percent.

ÖZET

PES PLANUS VE PES CAVUS İÇİN GÖRÜNTÜ İŞLEME VE DERİN SİNİR AĞLARINI KULLANAN MOBİL ON TEŞHİS ÇÖZÜMÜ

Bu alandaki önceki araştırmalara göre, dünya nüfusunun yaklaşık yüzde otuzunda bir çeşit ayak deformitesi vardır. Tüm deformiteler arasında, ayağın medial longitudinal arkının eksikliğinden kaynaklanan pes planus ve aşırı yüksek plantar longitudinal arkın oluşturduğu pes kavus, toplumun yaşam kalitesini en olumsuz etkileyen ayak deformiteleridir. Yukarıdakiler göz önünde bulundurularak önerilen çözüm, literatürde var olan geleneksel deformite tanıma yöntemleri yardımıyla görüntü işleme ve derin sinir ağları kullanılarak bir cep telefonu uygulaması aracılığıyla pes planus ve pes kavus ön teşhisini yapmayı amaçlamaktadır. Bu amaçla, bu çalışma, ayak deformitesi tespit hassasiyetini geliştirmek için birbiri üzerine inşa edilen üç çalışmadan oluşmaktadır. Buna göre, ilk prototip, pes planus ve pes kavusu tahmin etmek için makine öğrenimi teknikleri kullanılarak 34 katılımcı ile test edilmiştir. Bu prototipi kıyaslamak için bir ortopedistten, karşılaştırmalı bir hesaplamanın yapılabilmesi için kilit karar verme noktalarını manuel olarak sağlaması istenir. İkisinin karşılaştırma sonuçları, uzman ve prototip bulgularının yüzde 90'dan fazla birbiriyle uyumlu olduğunu göstermiştir. Bunu takiben, ikinci bir çalışma olarak, arka ayak açısı ve sağlık profesyonellerinden girdi içeren başka bir deformite sınıflandırma yöntemi uygulanmaktadır. Bu prototip 9 katılımcı ile test edilmiştir. İkinci çalışma kapsamında sağlık profesyonelleri tarafından prototip kullanılarak veri girişleri sağlanmıştır. Sistem sonuçları sağlık profesyonelleri tarafından tespit edilen deformiteler ile karşılaştırılmış ve sonuçların 27,7 oranında eşleştiği gözlemlenmiştir. Üçüncü ve son çalışmada ise birinci prototipte geliştirilen algılama yöntemi geliştirilmiş ve ikinci çalışmada kullanılan test seti ile test edilmiştir. Sonuçlar, hekim ve sistem tarafından tespit edilen deformitelerin yüzde 83,3 eşleştiğini göstermiştir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	iv
ABSTRACT.....	v
ÖZET.....	vi
TABLE OF CONTENTS.....	vii
LIST OF FIGURES.....	ix
LIST OF TABLES.....	xiii
LIST OF SYMBOLS/ABBREVIATIONS.....	xiv
1. INTRODUCTION.....	1
1.1. PROBLEM DEFINITION AND MOTIVATION.....	1
1.2. THESIS STRUCTURE.....	2
2. EVOLUTION OF TECHNOLOGY.....	3
2.1. ARTIFICIAL INTELLIGENCE.....	3
2.2. MACHINE LEARNING.....	4
2.2.1. Supervised Learning.....	5
2.2.2. Unsupervised Learning.....	9
2.2.3. Reinforcement Learning.....	11
2.3. IMAGE PROCESSING.....	12
2.3.1. Digital Image Representation.....	12
2.3.2. Binary, Grey-Level And Color Images.....	13
2.3.3. Image Processing And Filtering.....	14
2.3.4. Edge Detection.....	16
3. FOOT DEFORMITIES.....	18
3.1. PES PLANUS AND PES CAVUS.....	18
3.2. MEDICAL DETECTION SOLUTIONS.....	19
3.3. PES PLANUS AND PES CAVUS DETECTION TOOLS.....	21
4. METHODOLOGY.....	22
4.1. FOOT DEFORMITY DETECTION.....	22
4.2. DEEP LEARNING.....	27
5. STUDY I - FOOT DETECTION AND PRIMARY DIAGNOSIS.....	33
5.1. ANALYSIS AND DESIGN.....	33

5.1.1.	Prediction Module	35
5.1.2.	Fine-Tune Module	37
5.1.3.	Batch Process	39
5.1.4.	Foot Index Calculations	40
5.2.	IMPLEMENTATION.....	41
5.2.1.	Development Process.....	41
5.2.2.	End-User Application.....	42
5.2.3.	Specialist Interface.....	45
5.2.4.	Services And Batch Process	48
5.2.5.	Database.....	54
5.3.	TEST AND EVALUATION	56
6.	STUDY II - BACK SIDE DIAGNOSIS.....	61
6.1.	ANALYSIS AND DESIGN.....	61
6.1.1.	Prediction And Fine Tune Module	62
6.1.2.	Index Calculation	62
6.2.	IMPLEMENTATION.....	63
6.2.1.	End-User Application.....	63
6.2.2.	Services And Batch Process	64
6.3.	TEST AND EVALUATION	66
7.	STUDY III - INNER SIDE DIAGNOSIS IMPROVEMENT	69
7.1.	ANALYSIS AND DESIGN.....	69
7.2.	IMPLEMENTATION.....	70
7.3.	TEST AND EVALUATION	71
8.	CONCLUSIONS AND FUTURE WORK.....	73
	REFERENCES	75

LIST OF FIGURES

Figure 2.1. Turing test [1]	3
Figure 2.2. Machine learning types.....	5
Figure 2.3. Graphical illustration of bias and variance [2]	6
Figure 2.4. Decision tree.....	9
Figure 2.5. Phong reflection model [3].....	12
Figure 2.6. Electromagnetic spectrum [4]	13
Figure 3.1. Pes planus [5].....	18
Figure 3.2. Pes planus [6].....	19
Figure 3.3. EMED [7].....	21
Figure 4.1. Calcaneus inclination angle [8].....	22
Figure 4.2. Metatarsal declination angle [9].....	23
Figure 4.3. Meary's angle [10].....	23
Figure 4.4. Lateral talocalcaneal angle [10].....	23
Figure 4.5. Staheli index AB/CD [10]	24
Figure 4.6. Chippaux-Smiraka index i/s [10].....	24
Figure 4.7. Clarke's angle [11].....	25
Figure 4.8. Arch index E/D+E+F [11].....	26
Figure 4.9. Arch height index E/D+E+F [12].....	26
Figure 4.10. Rearfoot angle critical points [13].....	27
Figure 4.11. Artificial neural [14].....	28

Figure 4.12. Artificial neural network[15].....	29
Figure 4.13. Recurrent neural network [16].....	30
Figure 4.14. Convolutional neural network example [17]	30
Figure 5.1. Architecture diagram.....	33
Figure 5.2. Prediction and fine-tune modules.....	34
Figure 5.3. Prediction module sequence diagram.....	35
Figure 5.4. Fine-tune module diagram.....	37
Figure 5.5. Fine-Tune Module - Web Application - Activity Diagram.....	38
Figure 5.6. Fine-tune module web application activity diagram.....	39
Figure 5.7. Batch process activity diagram.....	39
Figure 5.8. Android (left) and iOS (right) application UI.....	42
Figure 5.9. User application validations (iOS)	43
Figure 5.10. User application onboarding (iOS).....	44
Figure 5.11. User application feedback (iOS).....	45
Figure 5.12. Web application patient information	46
Figure 5.13. Web application critical foot points on image.....	47
Figure 5.14. Web application foot deformity options	47
Figure 5.15. Web application critical point assistance.....	48
Figure 5.16. Web application information boxes.....	48
Figure 5.17. Deep learning test - DeepLab, FCN, YOLO, Skin, MIDAS (left to right).....	50
Figure 5.18. Batch process minimum bounding box (green line).....	50
Figure 5.19. Batch process perspective transformation	51

Figure 5.20. Batch Process - Sobel - Absolute Gradient X, Absolute Gradient Y, Magnitude Gradient, Directional Gradient - Left To Right.....	52
Figure 5.21. Batch process Sobel combined.....	52
Figure 5.22. Batch process polynomial function.....	53
Figure 5.23. General view of database.....	55
Figure 5.24. Database user group table.....	55
Figure 5.25. Database diagnosis group table.....	56
Figure 5.26. Database files group.....	57
Figure 5.27. Participants BMI type distribution based on CDC in study I.....	58
Figure 5.28. Critical point results on study I.....	58
Figure 5.29. Index error normal distribution in study I.....	59
Figure 5.30. Length error normal distribution in study I.....	59
Figure 6.1. Architecture diagram of the updated system.....	61
Figure 6.2. Minor changes in end-user application in Study I.....	63
Figure 6.3. Region of interest results in Study II.....	64
Figure 6.4. Calculated critical points for rearfoot angle (red dots).....	65
Figure 6.5. Participants BMI type distribution based on CDC in study II.....	67
Figure 6.6. Foot deformity type distribution in study II.....	67
Figure 6.7. Rear foot angle vs foot deformity in automated process results.....	68
Figure 7.1. Accuracy improvement with polynomial regression window in study III.....	70
Figure 7.2. Greedy noise reduce improvement in study III.....	71
Figure 7.3. Slope calculation for error detection mechanism in study III.....	71

Figure 7.4. Foot deformity distribution in automated process results in study III.....72

Figure 7.5. Foot deformity and calculated slope in automated process results in Study III72



LIST OF TABLES

Table 2.1. 3x3 Kernel (box blur).....	14
Table 5.1. Participant statistics in study I.....	57
Table 6.1. Participant statistics in study II.....	66



LIST OF SYMBOLS/ABBREVIATIONS

API	Application programming interface
BMI	Body mass index
CDC	Centers for disease control and prevention
GDPR	General data protection regulation
JWT	JSON web tokens
KVKK	Turkish data protection authority
Lidar	Light detection and ranging
MQ	Message queue
OMCS	Optical motion capture system
RGB	Red green blue
RDBMS	Relational database management system
HSL	Hue saturation lightness
RFC	Request for comments
SSL	Secure socket layer
UI	User interface

1. INTRODUCTION

In this section, pes planus and pes cavus deformations, which are foot deformation types, are explained, also an overview of the motivation of the study is presented, and finally the content of the thesis is detailed.

1.1. PROBLEM DEFINITION AND MOTIVATION

In our daily life, we often do not think about our physical extensions (e.g., foot). However, the feet play an important role in the integrity of the movement system, as they carry the whole body. Some scientists even describe feet as "hardworking workers of the body" because of the critical tasks they perform [18].

Many different deformities are encountered in the foot since birth of the individual such as pes cavus, pes planus, traumatic injuries and neurological lesions. Foot deformities affect gait and cause compensatory changes in other joints. In addition, knee, hip, and waist problems are also seen. Foot deformities disrupt the shoe fit and require special in-shoe orthoses. Due to improper pressure, pressure points cause skin problems ranging from calluses to skin ulcers.

Since it is estimated that about a third of the population has some form of foot deformity, it is crucially important to detect foot deformities. Even though pes planus and pes cavus are not the most common deformations in the population, they are the foot deformations that most affect the productivity of individuals. Therefore, diagnosing and treating pes planus and pes cavus is exceptionally important for society.

Pes planus is described as a decrease in the slope of the medial longitudinal arch. The prevalence of pes planus in the specific age groups varies between <1 percent and 28 percent [19,20,21,22]. On the other hand, pes cavus is described as an increase in the plantar longitudinal arch slope. In addition, pes cavus is relatively less common than pes planus [23].

The first objective of the research is to diagnose pes planus and pes cavus remotely via mobile

applications. However, remote diagnosis of pes planus and pes cavus is extremely difficult since there is no reference point in foot detection. Therefore, as first objective of this work, deep learning networks will be utilized to find the reference (e.g. foot) in images. The second objective of the study is to model the sole of the foot to provide a more accurate preliminary diagnosis. For this purpose, the function of sole of the foot will be calculated by using image processing and regression.

Within the framework of this research, answers will be sought to questions such as;

- Whether deep learning algorithms are sufficient in foot detection?
- Is it possible to create the sole function with image processing?
- Is remote detection of pes planus and pes cavus possible, and can it be a viable solution with the achieved detection ratio?

In the literature, there are many methods for detecting pes planus and pes cavus. These can be divided into two categories: non-anthropometric and anthropometric. Anthropometric methods require external tools such as a digital radiography system. Although the literature has proven that radiological imaging provides much better detection, however it is not a suitable method for remote detection. Therefore, this work has been carried out with non-anthropometric methods such as Arch Height Index and Rearfoot Angle.

1.2. THESIS STRUCTURE

Chapter 2 provides an introduction to the domains of artificial intelligence, machine learning and image processing in detail. Chapter 3 describes the medical treatments and detection methods of pes planus and pes cavus. Approaches used to detect pes planus and pes cavus are discussed in Chapter 4. Chapter 5 provides an overview of data collection and detection system first, followed by a discussion of testing and results. Chapter 6 works on completely different pre-diagnosis algorithms and participants. Chapter 7, which includes discusses the improvement of the initial pre-diagnosis algorithm and its results. In the last chapter, potential improvements and new concepts for future work are mentioned.

2. EVOLUTION OF TECHNOLOGY

This chapter provides an overview of artificial intelligence, machine learning, and image processing.

2.1. ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI) is the ability of a computer to perform various activities similar to an intelligent entity. In order to understand whether these various activities are carried out by artificial intelligence, humans need observation. There are two different alternatives to observing intelligence. The first is introspection through the observation of thoughts, and the second is the observation of behavior, which is the result of intelligence activities.

In the 1930s, Alan Turing, a pioneer mathematician, took part in the construction of the first electronic computer. In 1950, he proposed the "Turing Test" to answer whether machines can think, which is shown in Figure 2.1.

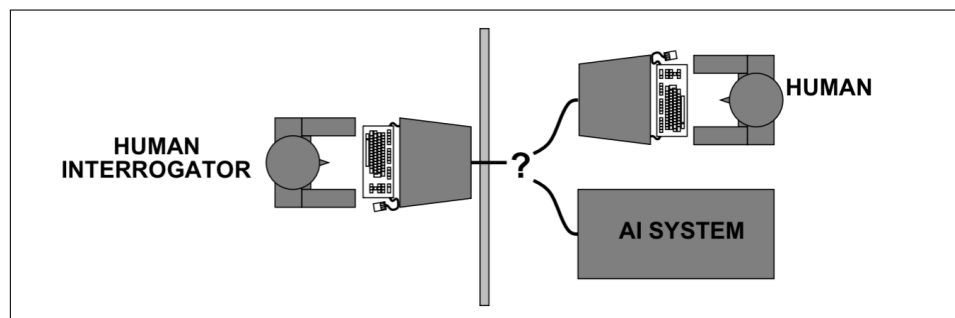


Figure 2.1. Turing test [1]

In the Turing Test [24], the subject communicates with the interrogator via a terminal. If the interrogator does not understand whether the subject is human or a machine, the subject is considered to have passed the Turing Test.

In the following years, programs such as SHRDLU [25], and ELIZA [26] were successful in the Turing Test even though they do not fully represent human intelligence. For example, the ELIZA program asks questions or forms sentences in response to the answers received from the user. Like a therapist, ELIZA asks open-ended questions. In addition, ELIZA forms the

questions according to the predefined words in the answer. Hence, when there is an input sentence that contains the word "friend," it would reply with "Tell me about your friend.". Thus, there is no understanding or representation of human intelligence. Therefore, based on statistical information, it is possible to pass the Turing Test.

On the other hand, John Searle, who argued that machines could never have the ability to think and understand, proposed the Chinese Room Experiment [27]. In the Chinese Room Experiment, an instruction manual tells what to do if specified Chinese letters. The person who is in the room writes in Chinese using this guide. Understanding Chinese is not replacing symbols with certain symbols. In this respect, he suggests that computer programs will never have the ability to understand.

There are three types of AI in the literature and; they are grouped based on their abilities. Weak AI (Narrow AI) [28] is the performing intelligent actions on a specific field (Narrow-Area) like the ability to drive a car. It is mainly emulating intelligence. General intelligence (Strong AI) [28] is the ability to perform human-like intelligence in various tasks. It has equivalent human behavior. Lastly, Super Intelligence [29] is the performing intelligent action that exceeds humans' performance in all domains.

Apart from this, there are many areas of AI such as Knowledge-based systems [30], Natural Language Processing [31], Fuzzy Logic [32], Robotics [33].

After a brief introduction to artificial intelligence, the following subsection will make an introduction to machine learning and machine learning types.

2.2. MACHINE LEARNING

Machine learning is the ability of machines to learn from experiences, make decisions regarding similar situations in the future, and produce solutions to problems [34]. Therefore, machine learning makes generalizations for a new related event by searching previous cases and outcomes [34].

There are three types of learning methods: Supervised, Unsupervised, and Reinforcement (Figure 2.2). The first one, Supervised Learning, requires an external intervention or an

internal mechanism to achieve the desired output values. The secondary, Unsupervised Learning, utilizes the analysis of the inputs without external intervention. Last but not least, Reinforced Learning, acquires knowledge by interacting with its surrounding environment and processing the data via a trial and failure approach [35].

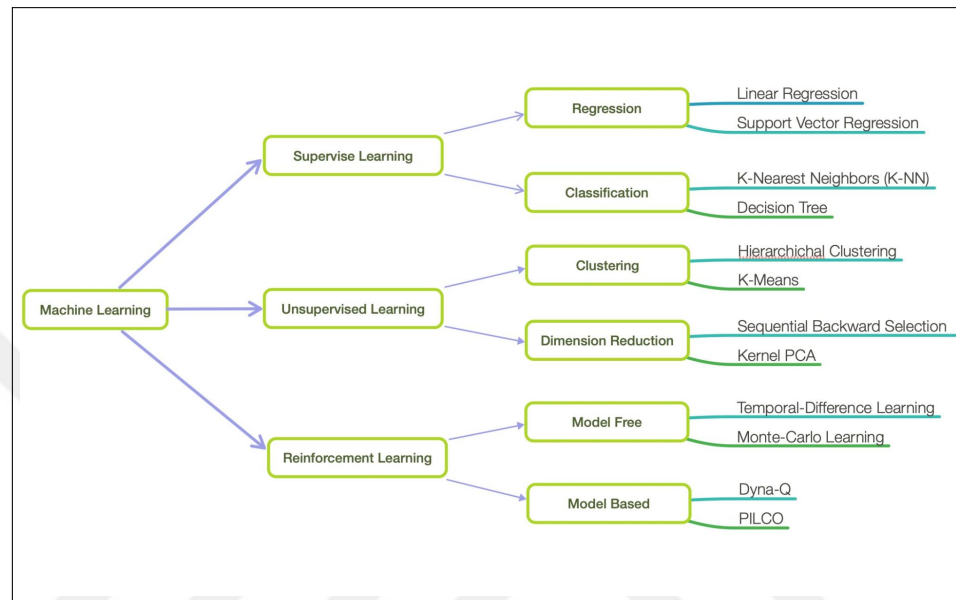


Figure 2.2. Machine learning types

2.2.1. Supervised Learning

In supervised learning, both the input and output values are fed to the algorithm in the learning phase. Therefore in the learning stage, the application processes input values and compares the desired output values with the result of the application. The difference between expected and output is treated as an error. The error value is calculated via a performance function. Result of the performance function is returned to the system to minimize this error. As a result, this process continues until the error is minimized.

In this error minimization, the algorithm generalizes outputs based on provided input data (i.e. Test Set) which is also called a supervised learning algorithm. This process could cause in four main problems. These will be discussed below.

The first one is the Bias-Variance trade-off [36] (Figure 2.3). A supervised learning algorithm has a high variance in predicting results with a significant deviation from the mean and other

predictions when training in various practice collections. Therefore, the variance measures the contiguity of the data predicted by the model to the actual data. On the other hand, supervised learning algorithms are highly biased if inaccurate when forecasting the output for different training sets. Therefore, the bias measures how incorrect the model is. As a result, the success of the algorithm is usually associated with the amount of bias and variance [37]. In most cases, the compromise between bias and variance is inevitable. In most of the algorithms, this trade-off can be modified with a parameter or automatically.

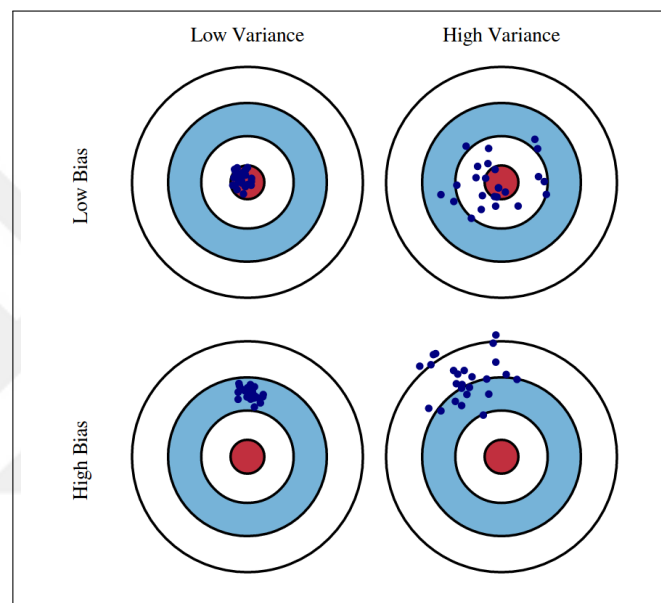


Figure 2.3. Graphical illustration of bias and variance [2]

The second problem is the complexity of the problem and the amount of training data. The amount of data required to train an algorithm depends on the complexity of the problem. In relatively simple problems, the small set of training data will be enough to achieve learning. However, complex problems, which require feature extraction in multiple dimensions, need a well-distributed and an extensive training set.

The third problem is the number of dimensions in an input space. The larger the number of dimensions, the more difficult the algorithm will be to learn because of the high variance, as redundant dimensions hide features. For this problem, there are algorithms to reduce dimensions like autoencoders. Another approach for this problem would be manually extracting features before feeding them to the algorithm. Alternative strategy would be fine-tuning the supervised learning algorithm to low variance, and high bias may also solve

the problem.

Apart from this, there is a possibility of overfitting in complex problems even if there is no noise. This type of overfitting is related to algorithms being unable to model the training data, also called deterministic noise. There are a few possible solutions to this problem. The first is removing data which caused overfitting from the data set by hand. Another approach is to use other algorithms to remove data that caused overfitting from possible errors from data sets and the last would be to use early stopping to prevent overfitting. For example, the drop-out layer is used in deep neural networks to prevent overfitting.

Supervised learning is used to tackle two types of problems: Regression and Classification. Regression is a collection of statistical techniques to predict the associations between variables. It tries to model and analyze multiple variables when concentrating on the connection between a dependent variable and one or more unrelated elements. Moreover, regression is used extensively for prediction and projection.

There are two types of regression. These are parametric and non-parametric. In parametric regression, the number of dimensions is predefined, such as Linear Regression and Ordinary Least Squares. On the other hand, non-parametric regression has no defined dimensions. As a result, regression models are often helpful in prediction, even if the assumptions are moderately wrong. Nevertheless, in many cases, regression models can not operate optimally. In the following paragraphs, we will examine a few Regression algorithms.

The linear regression method is a statistical technique that establishes a linear relationship between the independent variable and the dependent variable to estimate the feature variable [38]. In a linear regression equation, where an example is shown in Equation 2.1, x represents estimator for a problem, y is the response variable a the slope of the regression line is a_1 and, b_1 is the point at which the regression line intersects the ordinate axis.

$$y = a_1x + b_1x \quad (2.1)$$

The main working principle of the support vector machine (SVM) is to divide the data into the broadest categories that separate them from each other, which is based on

Vapnik–Chervonenkis theory developed by Vladimir Vapnik and Alexey Chervonenkis. SVM is successfully used in applications such as pattern recognition, time series analysis, and classification. SVM can be applied to both linear and non-linear data. The regression process is carried to the high dimensional feature spaces by the kernel function in non-linear data.

The other type in which supervised learning is used is a classification that assigns each input vector to a finite number of discrete categories [39]. One of the most known examples of this classification is classifying emails as spam or not. Beside that, machine learning studies for classification are used in various fields such as stock estimation, customer risk analysis for bank credit, earthquake prediction, text classification, disease diagnosis, genetic studies. In the following paragraphs, few classification algorithms were examined.

K-Nearest Neighbor Algorithm (K-NN) is a memory-based classification, where its popular use is in the data mining domain. In K-NN, after the classes of the sample set are determined, the method determines which new observation belongs to which class. Here, the samples are considered a point in n-dimensional space, and the parameter k, the number of neighbors closest to the given point, is determined [40]. The distances of all other points to the given point are calculated individually because of this method based on distance calculation.

$$d(x, y) = \sqrt{\sum_{j=1}^N (x_j - y_j)^2} \quad (2.2)$$

The Euclidean Distance shown in Equation 2.2 is commonly used to calculate the distance between observations. The smallest k is selected according to the distance values calculated here [41].

Another classification algorithm is decision trees. They consist of internal decision nodes and end leaves (Figure 2.4). A decision tree tries to establish which class the data with a particular class will be included in, starting from the node with the highest information gain [42]. One of the most commonly used algorithm to create a decision tree in data mining is the ID3. The main principle of the ID3 algorithm is to classify objects by testing the values of their attributes [43].

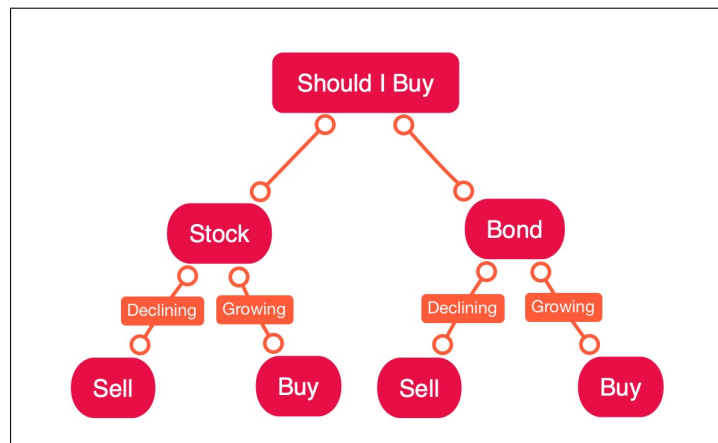


Figure 2.4. Decision tree

2.2.2. Unsupervised Learning

The purpose of unsupervised learning is to model the underlying structure or distribution of the data to extract information from raw data. Unlike Supervised Learning, there is no data to guide the learning process. It is generally used for clustering, finding relationships between attributes and size reduction. Besides, the results obtained with the unsupervised learning algorithm can be used for supervised learning [44].

A critical point for unsupervised learning is that, while the information transmitted to learning algorithms is highly rich in underlying composition, the objectives and prizes for training are usually very narrow. Therefore, in this type of objective and prize, the algorithm should understand the information instead of applying it to a specific task. As a result, using unsupervised learning as a sub-task might improve the results even in supervised learning tasks (e.g., Deep Learning Tasks).

Unsupervised learning is used for clustering and dimension reduction. Clustering is a data grouping with similar characteristics in a data set. Therefore, there are high similarities within the same cluster, and there are low similarities between clusters. K-Means and Hierarchical Clustering are some of the commonly used clustering algorithms. These algorithms are frequently used in areas such as customer segmentation, market segmentation, and computer vision.

Clustering is a very fundamental task to discover unknown elements of unlabeled data.

However, success measures depend on the users' demands, such as a minimum number of clusters. On the other hand, understanding the underlying knowledge in the cluster solely depends on the user. In the following paragraphs, we are examining K-Means and Hierarchical Clustering algorithms.

Hierarchical clustering (Hierarchical Cluster Analysis - HCA) is a cluster analysis technique designed to produce a cluster hierarchy. There are two types of HCA; namely Agglomerative (Bottom-Up) and Divisive (Top-Down). Agglomerative type starts by turning each data into a cluster, and then the clusters that are close to each other in distance form a new cluster. Divisive is the opposite of Agglomerative. At first, all data is created in a single set, then this cluster is fragmented and reclustered.

After the K value (the number of clusters) is determined by the user in K-Means, the algorithm selects random K points in the data set. Then, it calculates the distance between each point and random k-points to assign the data to a cluster. After assigning the points to clusters, k points are re-calculated based on the data in each cluster. Finally, the same distance calculation is performed according to the new k points. The method continues until a point is not assigned to relevant groups.

Real-life data has a large number of dimensions. This size of data brings complexity. This complexity also creates some problems; for example, there would be a significant increase in the amount of time and resources required to build a model. In addition, there is a high correlation between some attributes, leading to overfitting [45]. Dimensionality reduction, which decreases unrelated features or merges attributes into smaller sets of dimensions [46], helps us solve this problem. The following paragraphs examine the Kernel Principal Component Analysis (KPCA) and Sequential Feature Selection (SFS).

KPCA is based on standard deviation, covariance, eigenvectors, and eigenvalues. The use of integral operator kernel functions and non-linear maps can correctly compute main parts in high-dimensional feature spaces associated with input space [47].

SFA uses a greedy approach which is very useful for computationally complex problems. However, it might produce unoptimized solutions. SFA reduces or generates attributes at each iteration until the required dimension size k is reached.

2.2.3. Reinforcement Learning

Reinforcement learning is based on the reward system, which helps evaluate results coming from the environment by adapting itself to the actions it encounters [48]. Therefore, there is no testing and training phase in reinforcement learning. Instead, the reward system tries to give an insight into the environment by evaluating the outputs produced by the system as good or bad.

Reinforcement learning has many applications, from robot control to simple board games. For example, a chessboard game contains simple rules, but the opponent's strategy is difficult to predict in advance because of possible moves. Therefore, the system trained with the reinforcement learning model evaluates each move of the opposing player and tries to make the best move against his opponent. Accordingly, the system anticipates possible moves and takes a position accordingly [49].

Reinforcement learning can be performed both model-based and model-free. In model-based, algorithms try to predict the dynamics of the environment, such as alpha zero. On the other hand, model-free algorithms try to optimize the rewards without modeling the environment, such as q-learning.

The most well-known among the model-free reinforcement learning approaches is the Q-learning method. It is often applied to maze and search problems [50,51,52]. Watkins, who first proposed this method in 1989, used the letter Q for the value function where, the method got this name [53].

There also many model-based reinforcement learning approaches such as World Models [54], I2A [55], MBMF [56] PILCO [57]. PILCO is one of the most popular model-based reinforcement learning algorithms. Unlike other model-based models, it uses Gaussian Process for function approximation.

2.3. IMAGE PROCESSING

In digital image processing, analysis, and computer vision have made significant advances in theory and applications in recent years. Digital image processing has been a technological leader in developments in many vital areas such as medical view multimedia systems, biology, metal science, robotics and manufacturing, intelligent sensing systems, remote sensing, graphic arts, and printers [58].

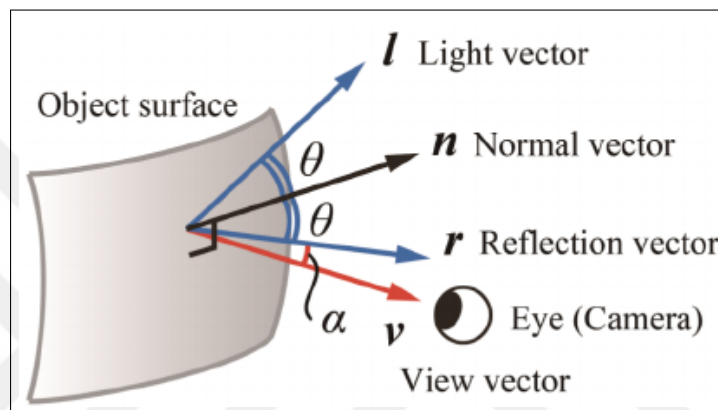


Figure 2.5. Phong reflection model [3]

Digital image processing techniques consist of 3 different classes; these are low-level vision, mid-level vision, and high-level vision. Low-level vision algorithms are specifically digital image processing algorithms such as edges detection: their input and output are digital images. Mid-level vision algorithms consist of digital images as input and image properties as output such as object recognition, 3D reconstruction etc. High-level vision algorithms, on the other hand, uses the higher level of perception such as conceptual description of activity, intention and behavior [58,59]. In the rest of this section, information about Low-Level Vision algorithms will be examined in detail.

2.3.1. Digital Image Representation

An image is the visual representation of an object by the capturing luminance emitted from the source. The mathematical model underlying image formation is based on the emitting source (e.g., visible light, X-rays, ultrasonic rays), the physics of the interaction of the emitting object, and the system is used (see Figure 2.5). Such a system usually includes an optical

lens, optical sensor, and image digitizer [58].

Digital images are generated by hardware that measures radiated energy from the source. These measurements are taken from various points across two-dimensional space to form the image. These measuring equipment are called sensors, and many different types are used. For example, sensors can interact with various parts of the electromagnetic spectrum, acoustic energy, electron emission, lasers, or other measurable signals. The electromagnetic spectrum includes visible light, infrared, ultraviolet, X-rays, microwaves, radio waves, or gamma rays (see Figure 2.6) [6].

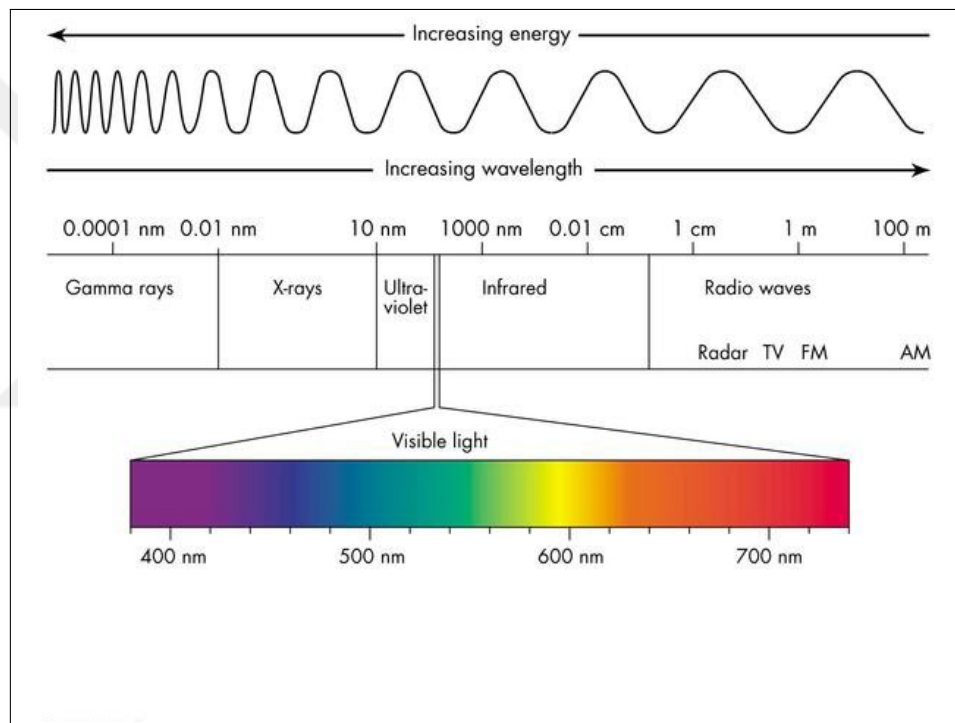


Figure 2.6. Electromagnetic spectrum [4]

2.3.2. Binary, Grey-Level And Color Images

Binary images are the simplest form of imagery which can be typically represented as "black" and "white" or "0" and "1". A binary image assigns 1-bit to each pixel in the image which is enough to represent each pixel [60,61]. In addition, binary images are often produced according to a defined threshold value from gray-level images. Here, each pixel above the threshold value gets the value "1", and the one below the threshold value "0" [60,61].

Gray-level images do not contain color information and only describe luminance information, also referred to as monochrome images. The number of bits used for each pixel is defined by the number of different brightness levels available. For example, a typical image contains 8-bits per pixel, which allows image to have 256 (0-255) different brightness (gray) levels [60,61].

Color images are expressed using color models that contain three or more values for all pixels. Therefore, at least three two-dimensional matrices should express color images. Some of the most popular color display systems are RGB (Red, Green, Blue), CMY (greenish blue, magenta, yellow), and HIS (hue, vibrancy, intensity) [61]. If the 8-bit monochrome image is used as the standard model, the corresponding color image should be 24-bit, with 8 bits for all color bands (Red, green, blue) for each pixel [60].

2.3.3. Image Processing And Filtering

Image filtering is the process of adjusting the pixels in an image to change its look. Filters can be used to increase contrast and provide a range of special effects to images [61].

Local processing is often required for the analysis of images. Since the character structure of the image to be processed can vary significantly from one region to another, it is natural that local processing is necessary. Many image processes can produce output information depending on the neighbors of the pixels of the image given as input information. In reality, there are many approaches based on this idea, often using sliding windows and varying from convolution to mathematical structure [61].

The kernel is a small matrix used in the convolutions of the image. Kernels of different sizes covering different patterns of numbers lead to different results under convolution. For example, Table 2.1 shows a 3x3 kernel presenting an average filter.

Table 2.1. 3x3 Kernel (box blur)

1	1	1
1	1	1
1	1	1

Convolution is applied by shifting the kernel on the image, usually starting from the upper left corner of the image. This operation takes place in all positions where the kernel will be applied, within the borders of the whole image, by moving the kernel. This application differs on the edges of the images. Each core position corresponds to a single pixel output calculated by multiplying the kernel values with the pixel value of the image and the sum of all these numbers together [62]. Mathematically expressed formula of the convolution can be seen in 2.3 where ξ is the filter kernel and $f(x, y)$ is the image.

$$d(x, y) = \xi * f(x, y) = \sum_{dx=-a}^a \sum_{dy=-b}^b \xi(dx, dy) f(x + dx, y + dy) \quad (2.3)$$

Convolution can be implemented with different kernels for various purposes such as blurring or edge detection, such as the Gaussian filter and the Sobel Edge Detection Operators. A few of the popular filters will be discussed in the coming paragraphs.

The box blur is replaces each pixel value of an image with the average value, including its neighbors and itself. Therefore, this operation results in the disappearance of pixel values that are not representative of their surroundings. This type of filtering reduces the relative differences between the neighboring pixels of the image and improves contrast by reducing noise and smoothing.

The second is the Gaussian filter, a convolution operator used to remove detail and noise and blur images. Therefore, it performs similar result to the box blur, but the Gaussian operator uses a different kernel, represented as a bell [63] (see Equation 2.4)

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2} \quad (2.4)$$

The effect of the Gaussian operator is to blur the image, as in the box blur filter. The Gaussian standard deviation determines the degree of blurring. Kernels used for a Gaussian with a more significant standard deviation will have larger dimensions. It produces the output by averaging the Gaussian center in the direction of the pixels. Therefore, it performs a smoother operation relative to the box blur filter and reveals better edges than a Gaussian similar-sized the box blur filter [63].

The last one is the median filter frequently used to reduce the noise in the images. As in the box blur, the median filter evaluates the pixels in the image together with the neighboring pixels. However, instead of taking the neighboring pixel values' average, it takes the neighboring pixel values' median and replaces the pixel value [62].

2.3.4. Edge Detection

Edges are places in the image that have significant brightness changes. Therefore, edge detection is widely used to detect the object boundaries that usually occur in the image regions. In addition, creating the image with the edges of the image can significantly reduce the amount of data while retaining most of the image information [62].

Since the edges are mainly composed of high frequencies, the edges can be detected by applying a high-pass frequency filter in the Fourier region or by convolution the image with a kernel. However, edges are detected by applying convolution with a specific kernel in practice because it is computationally inexpensive and gives better results [62].

The edges can also be found by calculating the derivatives of the image since they correspond to substantial illuminance. Therefore the edges can be detected by the maximum of the first derivative and the zero of the second derivative.

Different edge detection kernels make it possible to calculate an image's first and second derivatives. For example, Prewitt Compass Edge Detection [64], and Gradient Edge Detection operators [65] are two common approaches for calculating the first derivative of an image.

Prewitt Compass Edge Detection involves convolution of the image by adjusting the kernel sensitivity to identify a different edge. It determines the kernel edge size and definition that produces the maximum response at a pixel location. On the other hand, Gradient Edge Detection is a second and more widely used technique. Here, the image is convoluted with only two kernels for the x-direction and y-direction. The Sobel, Roberts Cross, and Prewitt operators are the most common kernels used as gradient edge detection detectors.

In Gradient Edge Detection, the pixels related to the edge need to be defined after calculating the magnitude of the first derivative. The simplest way is to set a threshold and consider

all pixels with a local value above the threshold to display edges. An alternative technique to a gradient image is to identify one pixel-wide edge at the local maximum point, which is a more common technique used by the Canny edge detection method. Canny first applies Gradient edge detection to the image and then finds the edge pixels by cutting non-maximum points and tracking hysteresis [66].



3. FOOT DEFORMITIES

Correspondingly, this chapter gives an overview of significant and relevant research in pes planus and pes cavus medical treatments and detection methods.

3.1. PES PLANUS AND PES CAVUS

Abnormalities and irregularities in the normal structure of the medial longitudinal arch cause functionally unstable conditions of the foot such as pes cavus or pes planus. Pes planus is the loss of the medial longitudinal arch of the foot that results in the entire bottom of the foot approaching or touching the ground while walking or standing [67] (see Figure 3.1). This arch acts as a flexible and adaptable foundation for the entire body [68], and its functionality is also vital to the body as it relieves the stresses of weight-bearing during the walking cycle and stores mechanical energy within the stretched elastic ligaments [69]. Although it may be asymptomatic, a defective medial longitudinal arch can alter the biomechanics of the lower extremities and lumbar spine, resulting in an increased risk of pain and injury [70].

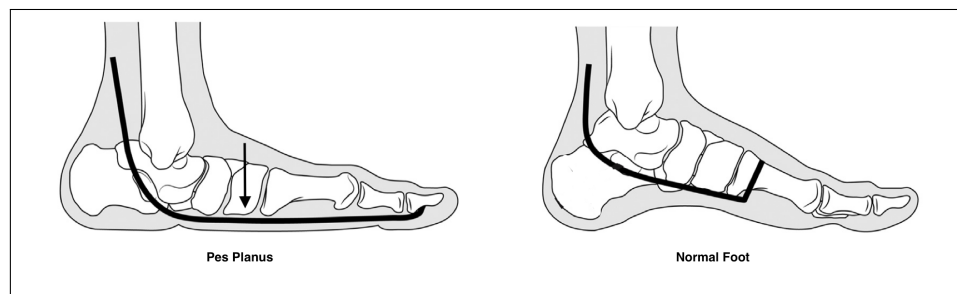


Figure 3.1. Pes planus [5]

Pes planus is caused by a variety of reasons that might be acquired or congenital. By the age of six, the majority of congenital pes planus abnormalities have vanished, and this is a normal aspect of human body development [71]. However, some congenital conditions can persist throughout adulthood, particularly those associated with obesity [72]. Pes planus can, however, develop as a result of various malformations in the extremities. One of the most prevalent causes of pes planus is functional issues with the posterior tibial tendon, which sustains the arch of the foot and allows for inversion and plantarflexion. Females over forty who have diabetes or are overweight are more likely to have posterior tibial tendon damage

[73]. In addition, pes planus can be developed more commonly in patients with congenital ligamentous laxity secondary to Down syndrome, Ehlers Danos or Marfan. Pes planus is also common among patients who have had midfoot or hindfoot trauma, such as a navicular, first metatarsal, or calcaneal fracture, or a Lisfranc ligament complex injury.

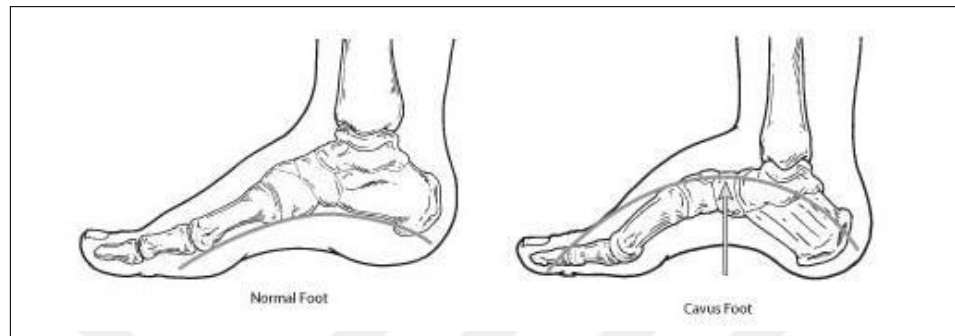


Figure 3.2. Pes planus [6]

Pes cavus, on the other side, is foot deformity distinguished by an elevation of the plantar longitudinal arch (see Figure 3.2). Clawing of the toes, posterior hindfoot deformity, plantar fascia contracture, and defects in the great toe are all common deformities associated with pes cavus. Furthermore, a common symptom of an underlying neurological disorder is pes cavus [74].

3.2. MEDICAL DETECTION SOLUTIONS

Pes planus and pes cavus have been diagnosed using ultrasonographic reviews, radiological evaluation, somatometric measurement, and clinical examination. In addition, indirect (or non-anthropometric) measurements are also utilized in the literature to detect pes planus and pes cavus, such as Inked or digital footprints (pressure measurements) and photographic [70,75]. However, radiological evaluations [76,77] are the most well-accepted and used approach for detecting pes planus and pes cavus in a wide range of solutions.

Chung et al. [78] used the well-accepted radiographic detection technique to evaluate radiological data of 103 subjects' navicular and talar heights and the arch index. According to their findings, the Staheli arch index, Chippaux-Smirak index, and Clarke's angle have 85.43 percent, 90.54 percent, and 83.89 percent prediction probability in preschool-aged children correspondingly. In another research, Pauk et al. [79] evaluate Clarke angle and radiography

data of sixty kids. Correlation between radiography and the footprints discovered in the research. Correspondingly, many research have been conducted to examine the relationship between footprint and radiography methodologies [80,81,82].

Aside from the comparisons research presented above, several studies in the literature only use non-anthropometric methods to detect pes planus or pes cavus, such as planimeter [83] or arch index [84,85]. Aside from the comparisons research presented above, several studies in the literature only use non-anthropometric methods to detect pes planus or pes cavus, such as planimeter [83] or arch index [84,85]. Consequently, the non-anthropometric methods' applicability has been demonstrated to be viable.

On 305 Maldivians between the ages of 13 and 17, Igbigbi et al. [85] used an arch index (footprint ratio) to identify arc type and pes planus ratio. Subjects' foot data is obtained in this study by making an imprint of the participant's sole with ink and a piece of paper. According to the authors, the technique employed is cost-effective, resilient, and more efficient.

However, Kanatli et al. [80] used 38 pre-schoolers and school-aged children with an average age of 6.4 (ages range from 3.7–11.7) to compare radiologic measures and footprint methodologies in order to determine the association between the two approaches. According to the data, there is a strong link between arch index, talo–horizontal angle, and talo-first metatarsal angle. On the other side, they discovered no link between arch index, lateral talocalcaneal angles, and calcaneal pitch.

A study that compared footprint and radiographic measures on 338 people found a decent association between the Staheli index, the Grivas Classification System, and the Chippaux-Smirak index with a similar impression [86]. The authors emphasize, however, that there is a limited association between the radiological measurement methods calcaneal pitch angles and talo-first metatarsal angle and all three footprint measurement methods as a result of the findings. However, the authors highlight a limited association between the radiological measurement methods, the talo-first metatarsal angle and calcaneal pitch angles, and all three footprint measurement methods as a result of the findings. Furthermore, the findings show that the footprint measurement methods and talo-horizontal angle have no meaningful correlation.

3.3. PES PLANUS AND PES CAVUS DETECTION TOOLS

There are a variety of produced systems for detecting foot abnormalities, ranging from simple solutions like planimeter [83] to far more advanced systems like gait analysis [87].

Gait analysis [88,89] is used by the majority of commercially available devices to detect a variety of foot abnormalities. Optical motion capture systems (OMCS) [90] are also used by some of these systems to detect foot abnormalities. Nevertheless, having this extra equipment for greater accuracy increases the required budget.

Moreover, some businesses like Sennotech's Senno Gait are designing low-cost solutions to address the market gap by reducing sensitivity. For example, Sennotech [88] analyzes gait using low-cost sensors and AI models. As a result, these devices are less expensive and easier to use, and they provide information such as injury concerns and irregular movement.

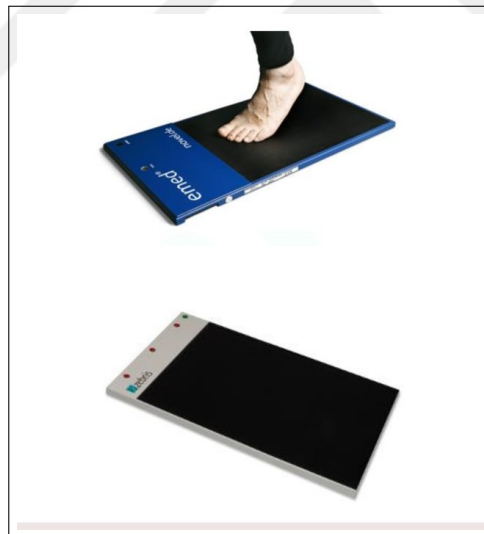


Figure 3.3. EMED [7]

Many researchers [91,92,93] have employed pressure detection devices [94,95]. These devices are far more widely available than OMCS equipment. However, they provide fewer data. Buldt et al. [91], for example, employed EMED (see Figure 3.3), a pressure sensing device, to assess pes planus and pes cavus in their research.

4. METHODOLOGY

This chapter discusses and details the methodologies in the literature that are used to detect pes cavus and pes planus, and as well as deep learning.

4.1. FOOT DEFORMITY DETECTION

One of the most efficient techniques to detect pes planus and pes cavus is using radiological scans. On the other hand, Pes planus and pes cavus may only be seen when the feet are pressed on the sole. Thereby, lateral radiographs are taken laterally or from the top of the knee and downwards for arch measures, while pushing the foot on the ground.

Many approaches for determining the state of the foot using radiological imaging have been documented in the literature. The use of calcaneal inclination angle, first metatarsal declination angle, lateral talocalcaneal angle, and Meary's angle are the most well-known of them all.



Figure 4.1. Calcaneus inclination angle [8]

The angle formed by a tangent line drawn between the lower face of the calcaneus to the ground is known as the calcaneus inclination angle (see Figure 4.1) [8]. If the calcaneal inclination angle is between 20 and 25 degrees, the foot is considered healthy. Pes planus, on the other hand, is defined as an angle of fewer than 15 degrees [96]. However, if the angle is more than 30 degrees, it is considered pes cavus [97].

The metatarsal declination angle is measured by considering the horizontal surface under

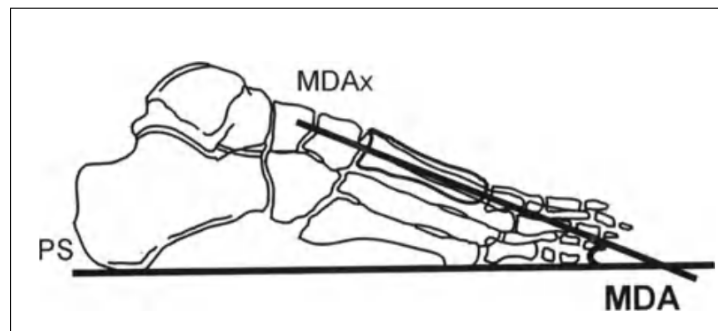


Figure 4.2. Metatarsal declination angle [9]

the sole and the calcaneal inclination axis and using a weight-bearing lateral foot radiograph (see Figure 4.2). The metatarsal declination angle in the general population is predicted to be around 21 degrees [33]. In any situation, the metatarsal declination angle is greater than 30 degrees, defined as pes planus [33].

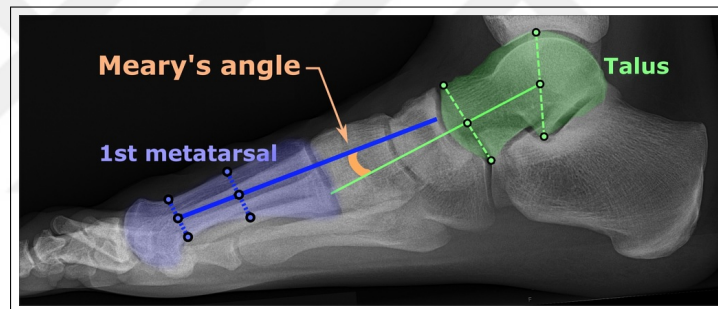


Figure 4.3. Meary's angle [10]

Meary's angle [8] is calculated by drawing a line through the centers of the longitudinal axes of the talus and first metatarsal (see Figure 4.3). The foot is accepted to be pes planus if the resulting angle is more than 4 degrees (convex downward) [98]. If the computed angle is less than -4 degrees (convex upward), the foot is classified as pes cavus [99].

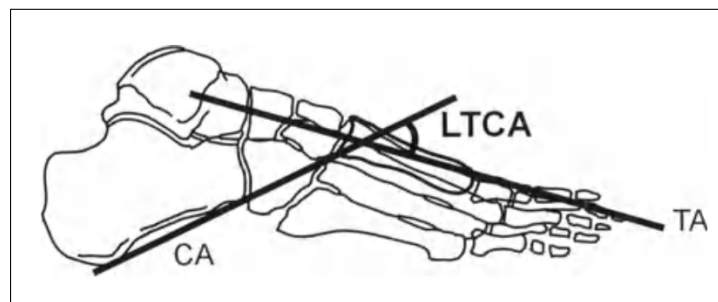


Figure 4.4. Lateral talocalcaneal angle [10]

The calcaneal axis and the collum lateral axis form the lateral talocalcaneal angle (see Figure 4.4) in a weight-bearing lateral foot radiograph. If the angle is less than 35 degrees, it is considered to be pes cavus. In contrast, if the angle is more than 50 degrees, it is diagnosed as pes cavus.

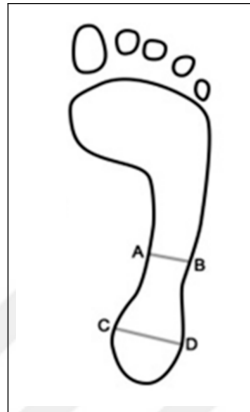


Figure 4.5. Staheli index AB/CD [10]

Compared to the anthropometric pes planus and cavus detection procedures stated above, non-anthropometric measurements, while less accurate, are significantly more readily available, less expensive to administer, and less harmful - i.e. do not require people to expose themselves to radiation. Therefore, Non-anthropometric procedures can be performed in large groups and in advance due to their ease of access.

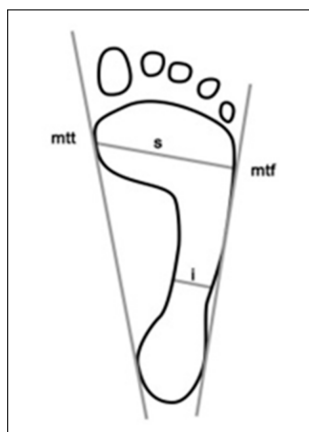


Figure 4.6. Chippaux-Smirak index i/s [10]

The footprint approach, which involves sinking the foot into ink and then pressing it onto graph paper, is one of the most common non-anthropometric methods used in the literature.

Many indexes, such as the Staheli arch and the Chippaux-Smirak indices, use graph paper to identify pes planus and pes cavus.

By dividing the width of a foot's center part by the width of the heel region, the Staheli index is determined (see Figure 4.5). The foot is considered to be pes planus if the determined ratio (index) is more than 0.8. However, if the estimated ratio is less than 0.4, it is classified as pes cavus [100].

On the other hand, the Chippaux-Smirak Index (see Figure 4.6) is calculated by taking into account the ratio of the midfoot's narrowest and widest regions. If the resulting proportion greater than 0.45 in the Chippaux-Smirak Index is defined as Pes planus. On the other hand, it is considered pes cavus [100] if the proportion is less than 0.25.

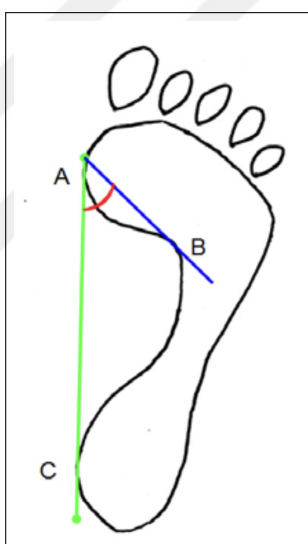


Figure 4.7. Clarke's angle [11]

The Clarke angle is formed by a line joining the tangent at the medial edge (A-C line in Figure 4.7) of the footprint and the longest vertical distance from the medial border of the foot to the point where the medial tangent intersects the edge of the forefoot (A-B line in Figure 4.7). Based on the size of the A and B lines' angle a foot is classified as, Normal (42° – 54°), mild flatfoot (35° – 41°), moderate flatfoot (30° – 34.9°), severe flatfoot (30°), and high arched foot ($> 54^{\circ}$) using the Clarke's angle approach [101].

The arch index is calculated by identifying the midfoot area (E in Figure 4.8) in ratio to the sum of the hindfoot (F in Figure 4.8), midfoot, and forefoot (D in Figure 4.8) areas. If the

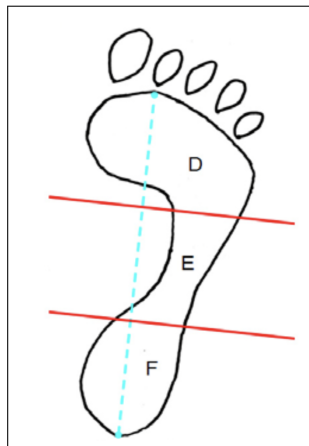


Figure 4.8. Arch index $E/D+E+F$ [11]

ratio of the arch index is equal to or larger than 0.26, the foot is considered to be flat. If it is less than 0.21, it is considered pes cavus. Otherwise, it is a normal arch [102].

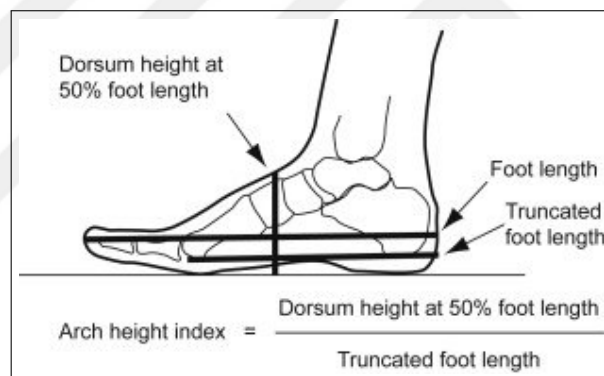


Figure 4.9. Arch height index $E/D+E+F$ [12]

There are other non-anthropometric procedures that do not require a footprint, such as arch height and rearfoot angle. Instead, these use intermediate results such as pictures to make measurements.

The arch height at 50 percent of the entire foot length is divided by the truncated foot length to calculate the arch height index (see Figure 4.9). In the arch height index average pes planus foot's ratio is expected to be 0.35 ± 0.03 and average pes cavus foot's ratio is expected to be 0.40 ± 0.03 [103].

Rearfoot angle is calculated using the angle between two lines, As illustrated in the figure 4.10. The lines contracted by the four points are the Achilles tendon center at the height of medial malleoli, the center of the shank posterior, 15 cm above the Achilles tendon center,



Figure 4.10. Rearfoot angle critical points [13]

and the base of the calcaneus [104]. If the calculated angle is between -4 and 4 degrees, the foot is classified as normal. On the other hand, If the computed angle is larger than 5 degrees and valgus, it is considered pes planus. If it is then 5 degrees and varus, it is classified as pes cavus [105].

4.2. DEEP LEARNING

Deep learning is a prominent object identification approach, a subset of machine learning that can be supervised (such as classification) or unsupervised. As a result, deep learning employs a large number of non-linear processing unit layers for feature extraction and conversion. Hence, each subsequent layer uses the output of the preceding layer as input and produces a classification [106].

One of the most influential aspects of deep learning is feature extraction. Low and high level (i.e. features extracted from low-level features) can be automatically extracted from data in deep learning. Therefore, manual feature extraction has become obsolete in deep learning.

However, deep learning algorithms require comprehensive data and high-performance hardware to process this data compared to other machine learning algorithms [107].

Deep learning is a very effective method that provides the ability to train and learn systems with large and complex probabilistic models. In order to achieve this, it uses different layers in a data representation. Therefore this provides the ability to pre-trained those layers separately. As a result, recent research shows that it provides more successful results than traditional approaches [108,109].

The data's size and complexity are significant factors in deep learning training. Therefore, increased complexity and size affect the training time. Transfer learning can be used to reduce training time by using deep learning algorithms trained in similar fields [106].

There are many deep learning architectures. Some of the significant architectures will be discussed in the following paragraphs starting with artificial neural networks then, recurrent neural networks, convolutional neural networks.

Artificial neural networks are a building block of deep learning that perform functions such as remembering, learning, generalizing, and producing new information by mimicking the human brain. Artificial neural networks are used in many fields and applications such as fingerprint recognition [110], autonomous vehicles [111], voice recognition [112], meteorological interpretation [113], handwriting recognition[114].

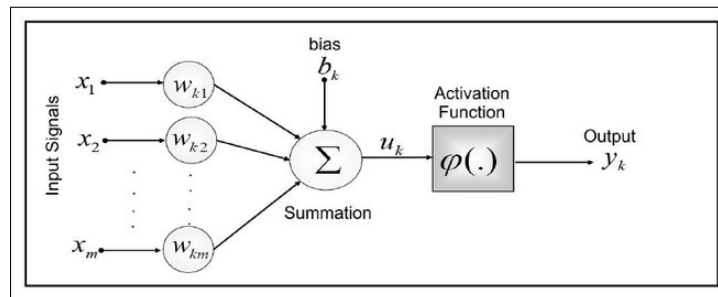


Figure 4.11. Artificial neural [14]

Artificial neurons have a straightforward structure. They consist of input, weights, bias, activation function, and output (see Figure 4.11). Inputs are the data coming into neurons from another artificial neuron or the initial input. The input is multiplied by a weight value and summed with the bias, which is optional. After calculating the summation, the output

is produced with the help of the activation function. In addition, each input effect can be adjusted with bias.

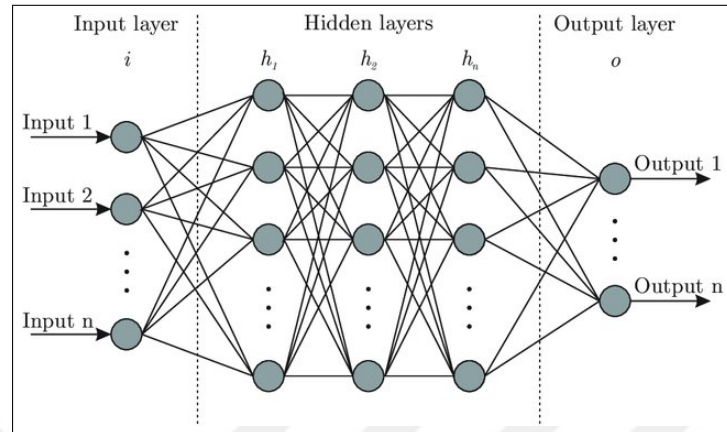


Figure 4.12. Artificial neural network[15]

The combination of artificial neuron cells form an artificial Neural Network. An Artificial Neural Network is made of three main layers. These are the input layer, hidden layer, and output layer (see Figure 4.12). The input layer brings the initial data into the system, which does not undertake any processing and relays that data directly to the lower layers. Next, the data is transferred into the hidden layer for further processing. Depending on the structure of artificial neural networks, the number of hidden layers may vary. Finally, after the processing, the output layer receives that processed data and offers the final results.

The calculated output and expected output values are processed according to the error function. Therefore, the error value is the difference between the calculated output and expected values. Finally, the calculated error value is updated using the optimization function. Various parameters, such as the number of data, the number of layers in the neural network, the activation function, and the learning rate [106], directly affect the success of artificial neural networks.

Recurrent neural networks (RNN) are another essential architecture with an internal memory structure. Their internal structure provides input history so that it correctly predicts the future. Whereas in conventional neural networks, all input data are assumed to be independent of each other, the basic idea of recurrent neural networks is to use sequential information [115]. Although RNNs are considered to be able to use information in very long sequences, in theory, it is known that returning to information is very limited in practice [115].

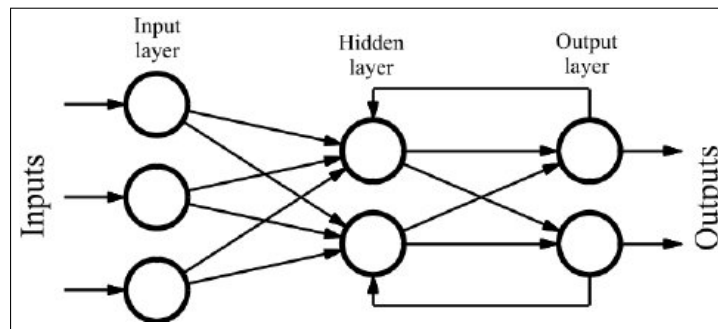


Figure 4.13. Recurrent neural network [16]

RNNs create a loop and use past data, which can be seen in Figure 4.13. RNNs are used in language modeling [116], machine translation [117], speech recognition[118], image description creation [119], and video tagging [120].

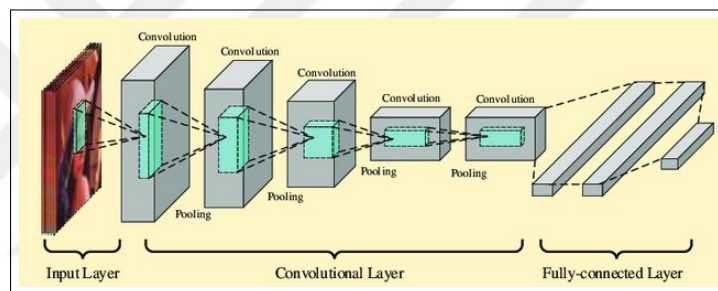


Figure 4.14. Convolutional neural network example [17]

Convolutional neural networks, which are created by taking as a model the human vision system, have achieved significant success in computer vision [121,122,123]. They are used in many areas such as object recognition, object classification, object tracking, sentence modeling. CNN may contain many layers: the input layer, convolution layer, ReLu, pooling layer, fully connected layer, dropout layer, classification layer, and output layer (see Figure 4.14).

The convolution layer, also known as the transform layer, is based on circulating a filter over all the input data. Filter sizes can be in different sizes such as 2x2, 3x3. The filters take the output data from the previous layer, apply the convolution operation and generate a feature map. Feature maps are regions where features specific to each filter are discovered. Edge information can also be found in this feature map [106]

The convolution layer, also known as the transform layer, circulates a filter over all the input

data. Filters can be in different sizes, such as 2×2 , 2×3 . These filters take the output data from the previous layer, apply the convolution operation and generate a feature map. Feature maps are regions where features specific to each filter are discovered. For example, edge information can also be found in this feature map [106].

The primary purpose of the pooling layer is to reduce the size of the input data. As the data leaves this layer, size reduction and information loss occur. This loss benefits the neural network as it reduces the computational load and prevents the system from memorizing [106]. Another layer that causes information loss is the dropout layer. Using this layer, some of the data is not transmitted to the next layer.

The fully connected layer is linked to all areas of the previous layers used in many network designs. Also, multidimensional data is converted to one dimension in the classification layer using the flattening method. As a result, classification is performed on one-dimensional data [106].

There are many object detection applications that have been created for the purposes such as people counting [124], autonomous vehicles [125], and face detection [126]. The general principle of object detection is to recognize a previously defined object class in a new image and define the positions of each object it detects in this image using a rectangle that encloses the entire object [106].

Many accurate and fast algorithms are available for object detection and tracking. However, the most straightforward deep learning approach for object detection and object tracking in deep learning is convolution neural networks. One of the disadvantages of this approach is that it does not recognize objects of different sizes. This requires each of them to be processed separately and requires a lot of computation time. Nevertheless, many neural networks are available for object detection, such as Fast R-CNN [127], Faster R-CNN [128], YOLO [129].

Semantic segmentation, one of the challenging problems in image classification, is defined as the labeling of raw pixels in an image with an object category. Deep learning approaches such as FCN and DeepLabV3 have been developed for semantic segmentation. For instance, an image is acquired using the Fully Convolutional Network (FCN) approach. Then, a segmented image of the same size as the input is produced. In this way, different probability

values can be obtained for each pixel in the input image [130]. DeepLabv3 is an approach built on top of the ResNet-101 [131] architecture by adding Atrous Spatial Pyramid Pooling (ASPP). ASPP applies convolution filters with different void ratios on the ResNet output to obtain feature maps with different details according to the void ratios of the filters [132].



5. STUDY I - FOOT DETECTION AND PRIMARY DIAGNOSIS

This chapter provides an overview of the primary system. Unlike other example solutions, the proposed approach will contain progressive updates with collected data over time. That is why the initial study contains fundamental structures to conduct the research. Accordingly, in Section 5.1, discusses general system analysis and design decisions. Section 5.2 covers detailed implementation of the prototype system, and lastly, Section 5.3 presents evaluation phase and initial test results.

5.1. ANALYSIS AND DESIGN

The system, when used by patients, calculates a preliminary detection of potential pes planus and pes cavus. Therefore, end-user interface, healthcare interface, back-end services, and database are needed. Each structure contains its own development life-cycle to provide the best results. The proposed structure of the system can be seen in Figure 5.1.

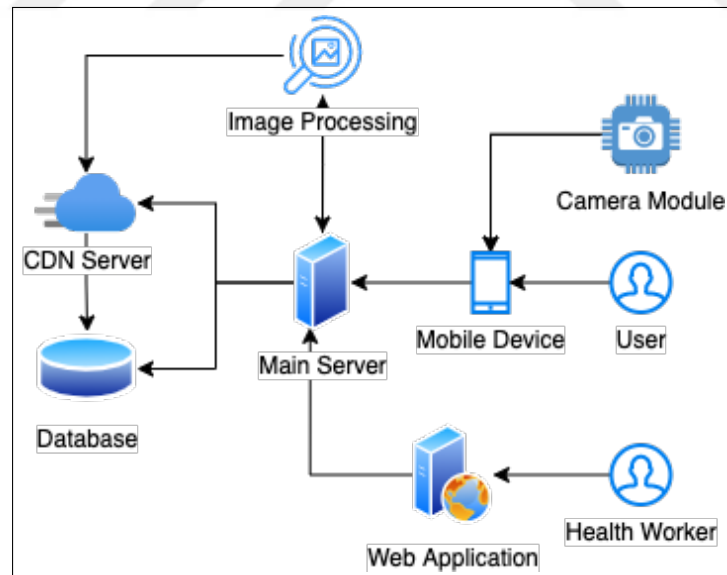


Figure 5.1. Architecture diagram

The goal of the study is to assist the medical staff in detecting pes planus and pes cavus and reduce the workload of medical staff. The system to be formed at the end of the study should provide basic functional requirements and as well as non-functional requirements, which are detailed below;

- Detection of pes planus and pes cavus with at most accuracy in provided images.
- Detects predefined extreme health cases and redirects users to healthcare professionals.
- Verify the results manually with an interface for experts
- View detection results
- Provide statistical reports about pes planus and pes cavus
- Provide the functionality to a wide range of users on multiple mobile platforms.
- Store all confidential information securely
- Store all essential information in non-volatile storage
- Design scalable system for large scale usage

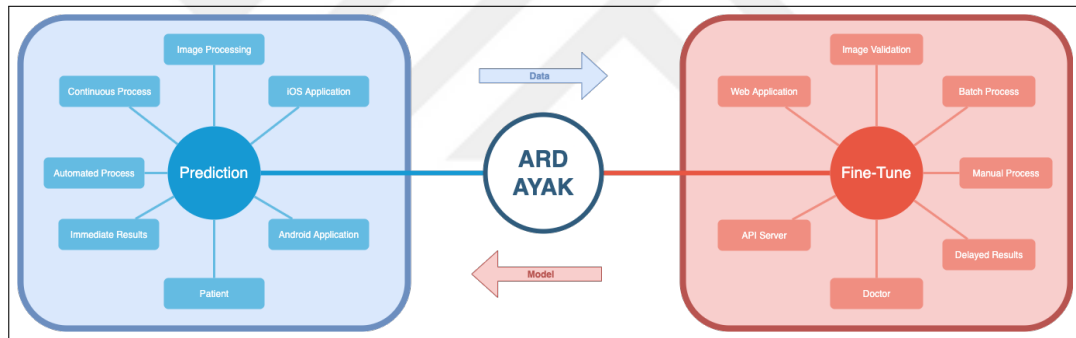


Figure 5.2. Prediction and fine-tune modules

Furthermore, all the services should not be accessible by unauthorized users since the system contains confidential information. Therefore, the same authorization process should be used in all information access. In addition, for research purposes, user data will be purified from user-identifying information when calculating measurement methods imposed by KVKK and GDPR, such as phone numbers.

The system should contain two essential components in order to provide a stable service: the prediction and fine-tune modules. In addition, modules should support each other and improve the process, as shown in Figure 5.2.

5.1.1. Prediction Module

The prediction module should essentially consist of two parts, namely end-user and pre-diagnosis application. The end-user application will provide remote access, which is one of the priorities of the system, to the user. Thus, patients will be provided with healthcare services without going to the hospital. To this end, mobile applications should provide services to 95 percent of the mobile operating systems in the market, which are mainly made of iOS and Android operating systems. After user information is collected through the end-user application, pre-diagnosis application will be used to diagnose the user and provide results.

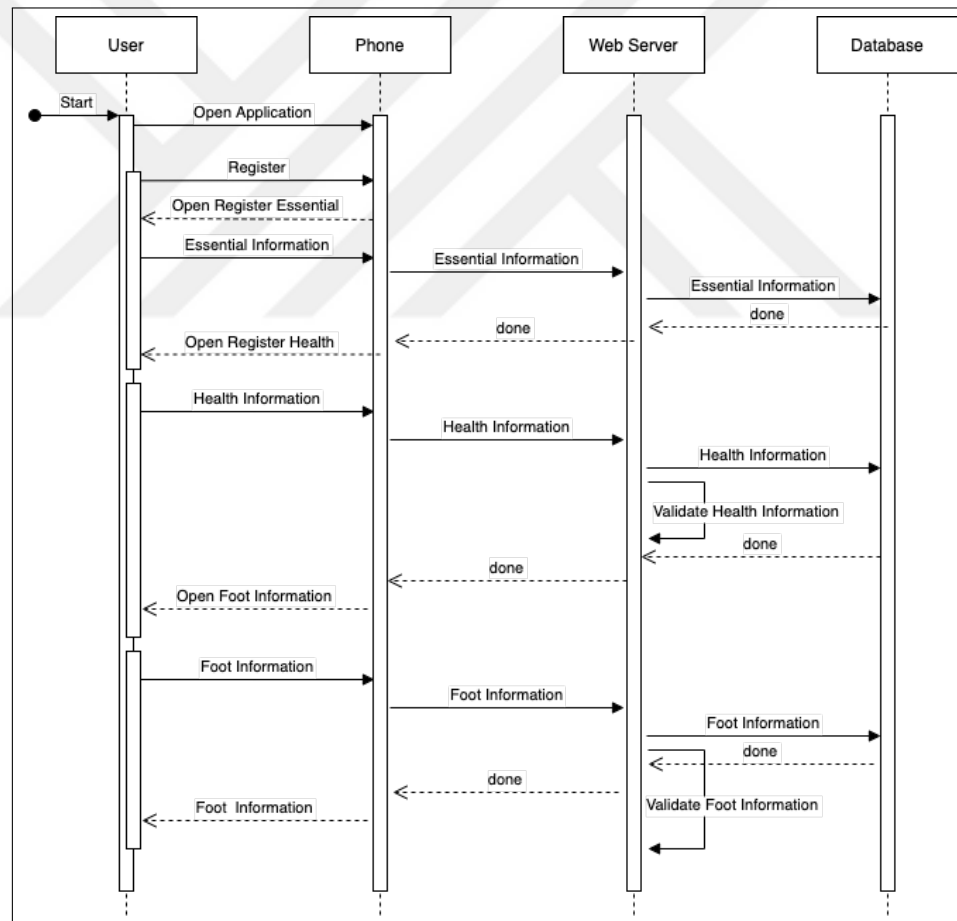


Figure 5.3. Prediction module sequence diagram

In the prediction module, user information collection is one of the fundamental parts. Information collection should be divided into three fundamental headings: essential, health, and foot information.

The essential information collection part focuses on differentiating end-users. Therefore, unique properties such as users' phone numbers or email addresses should be collected as part of this heading. Furthermore, gender, birth date, weight, and height should also be collected to analyze the population variance. Even if it is not related, the terms and conditions document that asks the user to confirm data usage and storage permission also should be viewed and agreed upon under this heading.

The second part is the health information collection which focus on detecting health issues related to pes planus and pes cavus. However, this part also requires the system to detect the evaluation requirement of health professionals for critical health issues. Therefore, the system should propose a direct call or appointment for the face-to-face examination between the user and health professional in such conditions. Furthermore, discovering critical health issues should be based on a rule-based system to reduce errors. However this is expected to be very limited to specified cases such as extreme pain levels.

Finally, the foot information collection part should focus on detecting pes planus and pes cavus. Therefore, the degree of pain in physical activities, the degree of pain at night, and most importantly, pictures of the foot should be collected. In Figure 5.3, the sequence diagram of the prediction module can be seen.

In the prediction module, the collected information should also inform health professionals in order to reduce examination time. Besides that, in Section 5.1.2, Fine Tune Module, the system performance improvement step, which is formed by adding new data to be obtained from health professionals, will be discussed.

In addition, all end-user applications should be user-friendly and provide sufficient feedback information for users. This feedback will improve user satisfaction and reduce the overall process's time consumption, such as application usage instructions.

Last but not least, the prediction module collects statistical data as part of the survey. Conducting surveys through the end-user applications will allow the system to reduce UI shortcomings. Apart from these, the same surveys will also enable healthcare professionals to receive feedback on actively performed healing procedures. Some procedures require users to perform actions on a daily or weekly basis. Therefore surveying these actions to improve

users' overall experience on digital healthcare is of utmost importance for the system to be more user friendly.

5.1.2. Fine-Tune Module

The fine-tuning module, which is the integral part, should include background procedures and healthcare professionals' interface. In order to achieve this, the module should implement two functionality namely system enhancement and user interface. Accordingly, in order to provide workflow, healthcare professionals, also known as system managers, should review and analyze user-provided data, which should be used to refine system diagnoses process. Eventually, the system should actively enhance the pre-diagnosis process by using the data doctored and adjusted by the system managers.

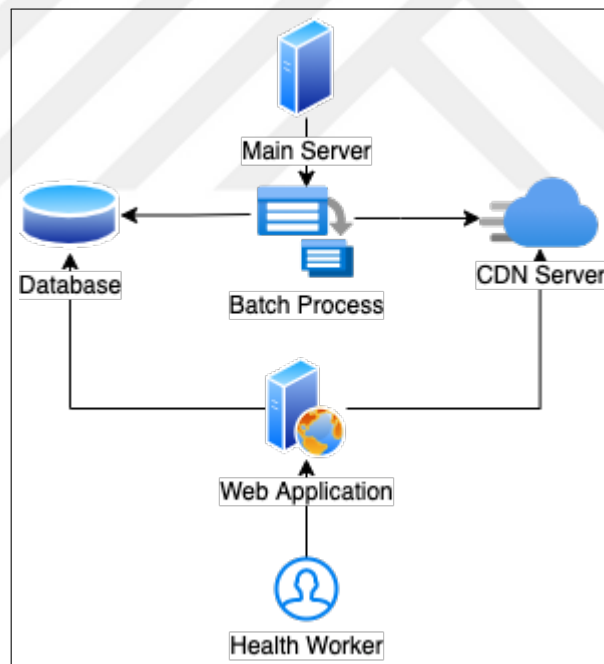


Figure 5.4. Fine-tune module diagram

As can be seen in Figure 5.4, a web application should be implemented for system managers to display and review the user input, such as foot images, previous diseases. In the web application, system managers should be able to calculate foot types with objective measurement methods such as the Staheli arch index with provided images from the user. As a result of this calculation, there will be two results, one provided from the doctor, and one

generated by the system which can be later compared.

System managers supply data in an orderly fashion in dual circles; provide, validate (see Figure 5.5). Healthcare professionals view patient information in the provided cycle and contribute foot type detection inputs. Information viewing involves primary data (BMI, Shoe Size), patient complaints (pain level, pain duration), health history (surgeries and chronic diseases), and foot pictures. Afterward, foot type detection inputs are provided by health workers. As a result of detection inputs, the system should be able to calculate multiple foot indexes. In Figure 5.6, the sequence diagram of the fine-tune module can be seen.

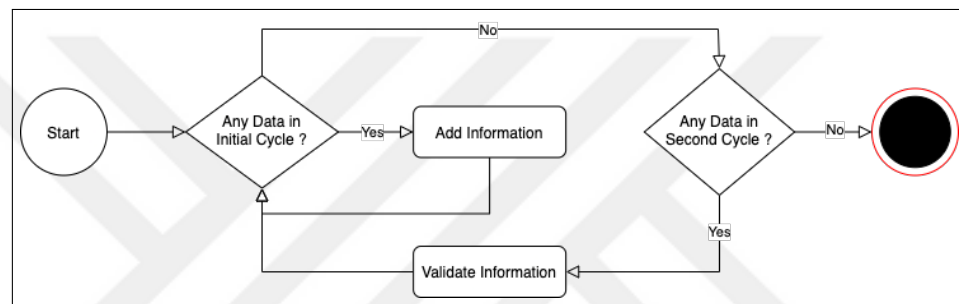


Figure 5.5. Fine-Tune Module - Web Application - Activity Diagram

At the end of the initial cycle, the system should calculate well known accepted foot indexes using key points obtained from healthcare professionals. These indexes should be pre-defined such as Chippaux-Smirak Index, Arch Height Index.

The second cycle should provide the calculated data to healthcare officials to validate the results. The system user should see how the results are calculated, related user data, and other meaningful information in this second cycle. At this stage healthcare professionals can respond to each of the results as move to initial cycle, unusable or approved. Accordingly, the approved results should be used to improve batch process performance

Furthermore, the second cycle should also focus on preventing any user-related errors. Therefore, multiple calculation results, formulas, and corresponding result statuses (pes planus and pes cavus) should be displayed to reduce the errors. In addition, the system should be able to rewind the cycle to the initial state is based on healthcare professionals' decisions.

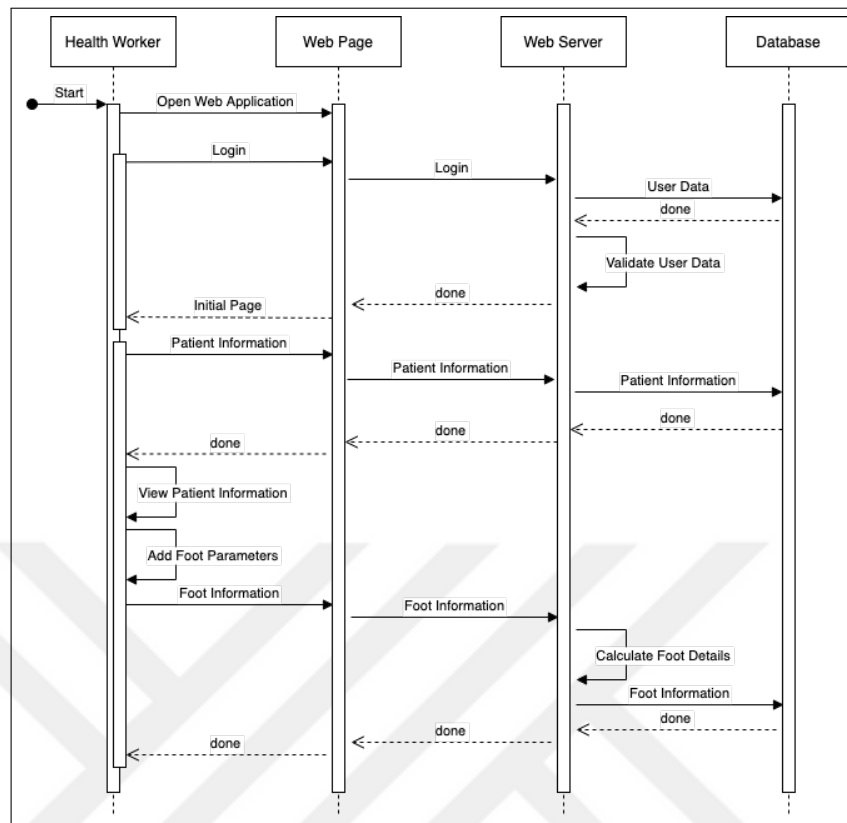


Figure 5.6. Fine-tune module web application activity diagram

5.1.3. Batch Process

The batch process should be designed to improve the pre-diagnosis with the data provided from the prediction and fine-tune module. In addition, the batch process should be automated for continuous learning. Therefore, this will provide the system to improve itself over time through the data provided by the modules (doctors and patients).

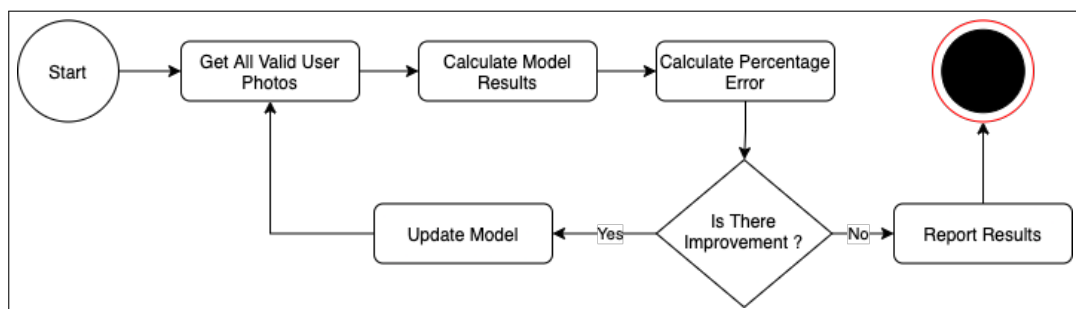


Figure 5.7. Batch process activity diagram

The batch process should consist of four basic steps. Firstly, the batch process should get all

the valid user-provided photos from the prediction module. Then, the system should process all the images on the current model and calculate the results. After the calculation, the batch process should calculate percentage errors with provided data from healthcare professionals. Then, based on results improvements, the model is updated or not. Finally, all the calculations should be redone when the model is updated until the best results are obtained. In Figure 5.7, the activity diagram of the batch process can be seen.

In addition, the batch process should provide reports about the system's detailed and overview progress, which contains calculated multiple errors. The overview results are a comparison of patients' diagnoses. As a result, this error calculation should provide how successful the system is as a product. Moreover, the detailed error results should be used to improve the system in short cycles. The detailed error calculation should use sub-findings of the system and the calculation points provided by health workers, such as points on the pictures used to identify foot structure.

5.1.4. Foot Index Calculations

The system should use multiple methods to find the best approach and assist healthcare professionals in diagnosing patients successfully. Therefore, a fine-tune module system should provide foot indexes, calculation methods used in the examination such as Arch Height Index, Chippaux-Smirak Index, Staheli Index, Clarkes Angle Calculation, Rearfoot Angle Calculation.

Each calculation type requires its specialty to diagnose accurately, such as the calculation of specific angles of the foot structure. Therefore, the system focuses on a specific type of index calculation, the Arch Height Index, in an automated process to reduce errors.

The arch height index should be calculated using a single foot image. Thus single image usage reduces user workload and improves user experience. In addition, the arch height index requires an image of the foot containing the side of the foot to calculate the index type correctly. This image should be used to detect reference points on the side of the foot.

Other index calculations are not used for the automated process. They were used only with the data provided by healthcare professionals, which details are explained in the offline phase

part. Instead, other index methods are used as reference methods for our calculations. This will allow us to compare different types of measurements. Consequently, the best method for the mobile approach will be selected. The calculation details can be seen in Chapter 4.

5.2. IMPLEMENTATION

This section discusses the implementation details and decisions of the study. The study contains multiple applications. Each application is divided into its implementation section. In addition, this section starts with the development process subsection, which explains applied development processes for each application.

5.2.1. Development Process

Software engineering is one of the fastest-growing fields in computer science. Over the years, it has had some milestones in development processes. However, in general, iterative development in its core has never changed. The most significant difference in software development processes is the reaction to changes in requirements. Therefore, a hybrid model of software development processes is applied, which are different in each application. This section will provide information regarding this decision and the reasons for it.

Client applications follow the prototype development process since collecting the data was crucial at the start of the development process. In addition, client applications contain dynamic composition where requirements change based on feedback. For example, only the iOS operating system was supported at the beginning of the project, but the Android prototype was also introduced due to the diverse mobile market and the users' need for Android applications.

On the other hand, batch processes, which used by the other application in Study I, were less prone to change and had no different types. Therefore, the waterfall development process is adopted. In addition, the server APIs for rapid response to client application changes have continuously improved throughout its lifetime as it fundamentally embraces Agile development.

5.2.2. End-User Application

The end-user system contains two distinct clients, which are iOS and Android. Client applications developed in mind usability, accessibility, inclusion, updated and improved during the application lifecycle. Consequently, this will improve the user experience and be satisfactory for all user groups.

The iOS client is compatible with iOS 11 and forward, making it accessible for 99,7 percent of the market share. Furthermore, mainly Swift programming language is used for the development of the client. Therefore, this makes it easy to support feature updates and compatibility for all the iOS ecosystems.

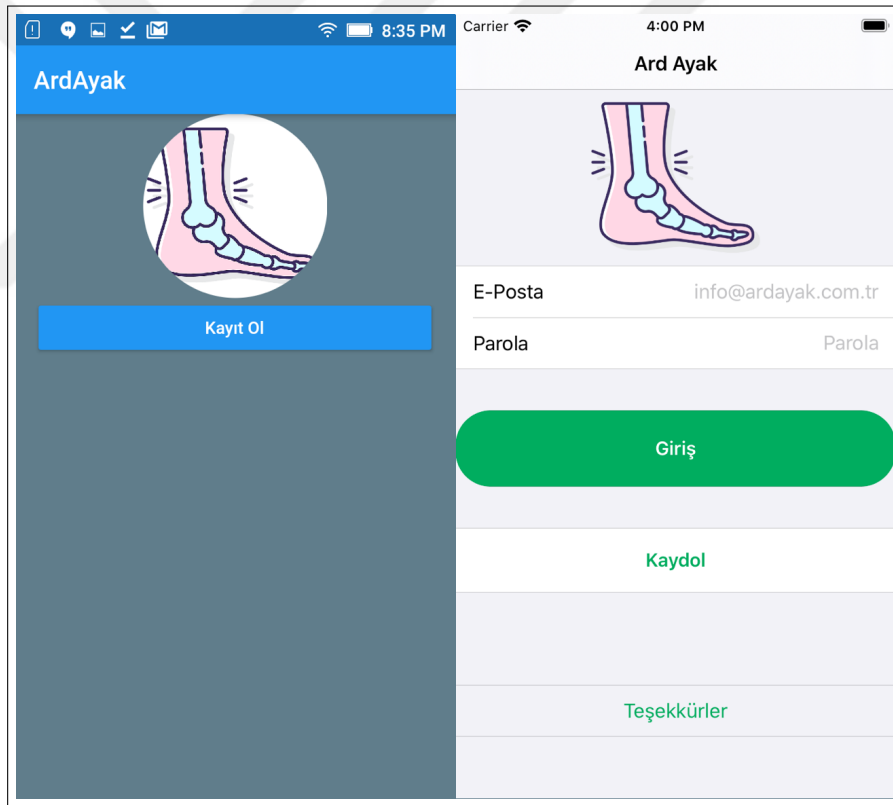


Figure 5.8. Android (left) and iOS (right) application UI

The Android client is compatible with API level 16 and forward, making it accessible for 99,8 percent of the market share. Furthermore, the Flutter software development kit is used in the development process, which requires Dart programming language. Therefore, this makes the software compatible with iOS and Android publishing. On the other hand, the Flutter application is only used for Android clients as this process requires additional

development. However, this feature can also be used in the future development process to reduce development costs.

Both end-user applications use the same user flow with corresponding platform UI specifications. Therefore this will enable users to complete their operations more comfortably with the UI they frequently use, shown in Figure 5.8. Furthermore, user documentation is also created to help smooth the user experience. This document contains all the information, from how to use it to the download links of the application.

Carrier 8:49 PM

< Ard Ayak User Information

Name
Name is required

Surname Test

Course

Student No

Phone Number 5XX XXX XX XX
Please, enter a valid phone number

E-Mail info@ardayak.com.tr
Please enter a valid email adress

Password j
Password should be min 8 max 13 char

Birth Date
Birth date is required.

Weight 0 kg
Your weight should be largen then 50 KG

Figure 5.9. User application validations (iOS)

Even though end-user documentation is created, client applications always consider user mistakes and misunderstandings. Therefore, input validation and feedback are implemented through the application. These validations include input type, max-min character, logical validations, and many more, shown in Figure 5.9, to improve users' experience. In addition, user tutorials are also added to guide the user, which is shown in Figure 5.10.

The image collection is a fundamentally similar process for both clients. This process includes taking a photo or taking it from the gallery. After that, since image sizes would be different in different hardware, iOS and Android prototype applications resizes the image and transfer it in

base64 format. Furthermore, file compression is used to reduce the traffic when transmitting the images from client to server. Deflate, a lossless compression format, is selected for compression since it is a widely used open-source compression algorithm available in major operating systems. In addition, there is a file format validation on clients for gallery-selected images.

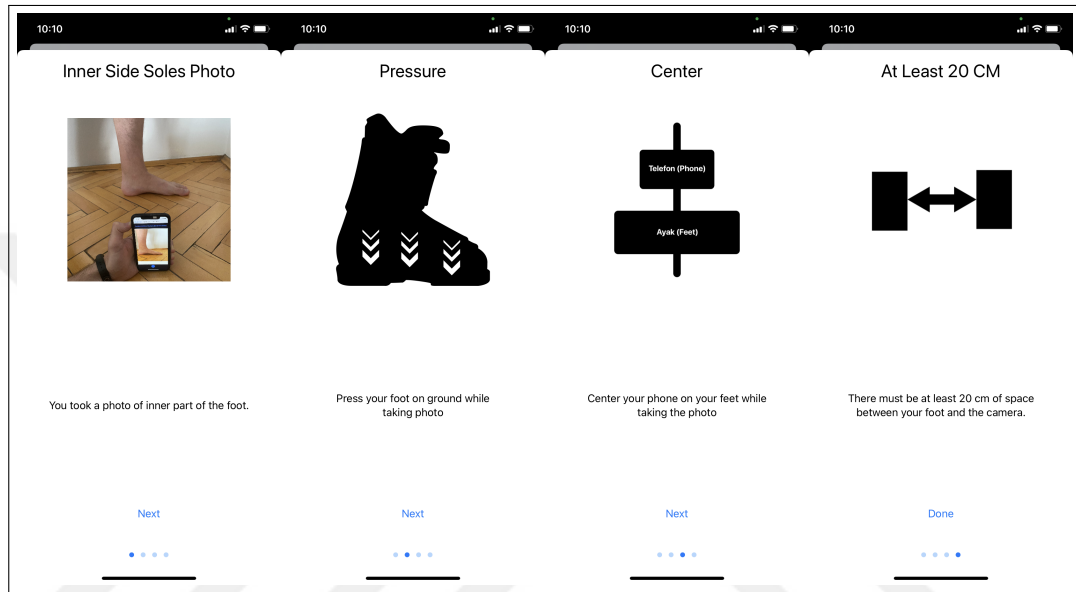


Figure 5.10. User application onboarding (iOS)

The shooting angle and feet pressuring the ground during the photos shooting are essential for pes planus and pes cavus detection. Therefore onboarding is displayed (see Figure 5.10) before each shooting. This onboarding shows a preview of how shooting should be and the state of the foot. As a result, this will reduce human-related errors furthermore.

Feedbacks are one of the most critical aspects of the user experience. Therefore, multiple feedback mechanisms are developed in client applications to reduce communication errors. For example, since image uploading relatively takes more time than the other operations in the application, a progress bar with percentages is developed (see Figure 5.11). In addition, success screens are developed to indicate to users that the process is completed (see Figure 5.11).

The most important difference between iOS and Android processors is that some mobile phone models provide information other than raw image data during photo capture. For example, some of the Apple hardware contains LIDAR sensors. Therefore, the iOS client

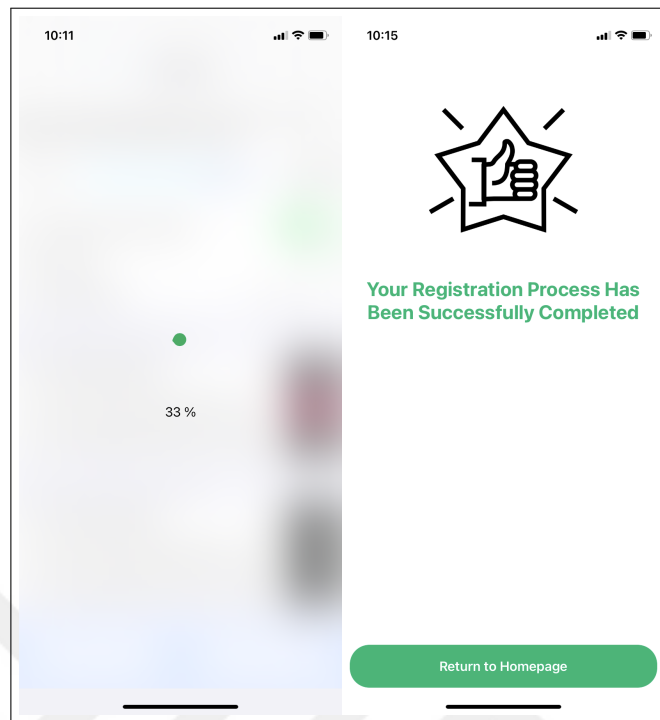


Figure 5.11. User application feedback (iOS)

uses this hardware advantage and sends extra information to the service, which is the depth data. This depth information could be used to increase the accuracy of pre-diagnosis. In addition, depth data is converted to Point Cloud Data format before being transferred to the server for future updates for other systems.

Lastly, localization is one of the features of the client applications. Applications support multiple languages to increase usability in multiple regions in the world. In addition, supporting other languages would be straightforward as adding a new file to the application language file base. Furthermore, the client applications use operating system APIs to open the application in users' preferred language.

5.2.3. Specialist Interface

The specialist interface is a web application whose primary purpose is to enable healthcare professionals to view patient data and diagnosis the patient based on the provided data. In the background, this process also improves the system by providing data to improve pre-diagnosis, which will be detailly discussed in the batch process.

Patient Information		Surgeries
User ID	25	No Surgery
Weight	54 kg	Chronic Diseases
Height	165 cm	
BMI	19.83 BMI	
Birth Date	9.2.1998	
Shoe Size	40 (~25 cm)	
-		No Chronic Diseases
Left - Pain With Activity	0	
Left - Pain Duration	0	
Left - Pain In Night	0	
Left - Pain In Morning	0	
-		
Right - Pain With Activity	0	
Right - Pain Duration	0	
Right - Pain In Night	0	
Right - Pain In Morning	0	

Figure 5.12. Web application patient information

The specialist interface is designed to work in two cycles. In the first cycle, the system provides essential patient information such as age, foot size, surgeries, and chronic disease, shown in Figure 5.12. After validating initial information, healthcare professionals can set key decision-making points in images provided by patients, which will be used to calculate novel-approved foot indexes.

The second cycle provides calculated results so that healthcare officials can make the final diagnosis of the patient's condition. Therefore, applications provide an overview of the patient information and display decision-making points in images to ensure they are correct (see Figure 5.13). Healthcare officials can redo the point set or diagnose the patient (see Figure 5.14). In addition, in case of unusable images, there is an option that asks the user to re-upload foot pictures.

The web application contains validations to ensure current diagnosis. Thus, the system ensures that all the critical points are set in the initial cycle. The second cycle provides that human-related errors are eliminated, such as wrongly placed points.

In addition, The web application also provides helper functionalities to healthcare workers

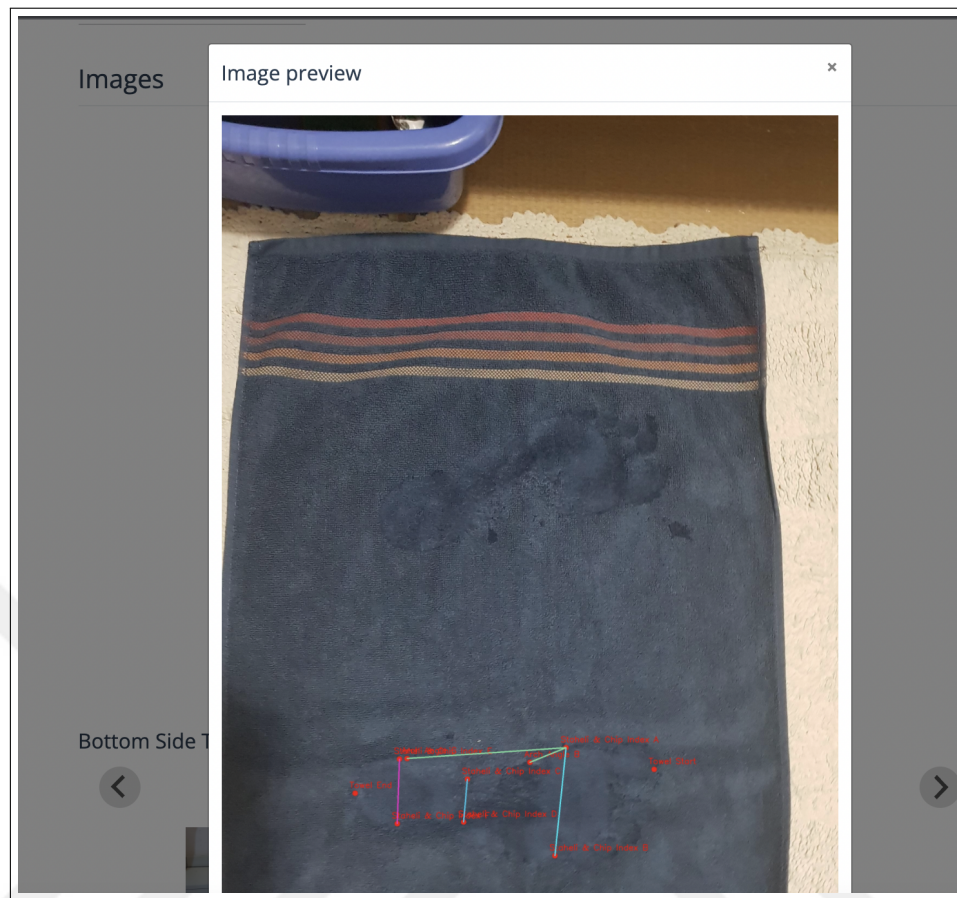


Figure 5.13. Web application critical foot points on image

Foot Deformity For Left Foot

☐ Healthy
 ☐ Pes Planus 1
 ☒ Pes Planus 2
 ☐ Pes Planus 3
 ☐ Pes Cavus
 ☐ Undetectable
 ☐ ToDo

Foot Deformity For Right Foot

☐ Healthy
 ☐ Pes Planus 1
 ☐ Pes Planus 2
 ☒ Pes Planus 3
 ☐ Pes Cavus
 ☐ Undetectable
 ☐ ToDo

Figure 5.14. Web application foot deformity options

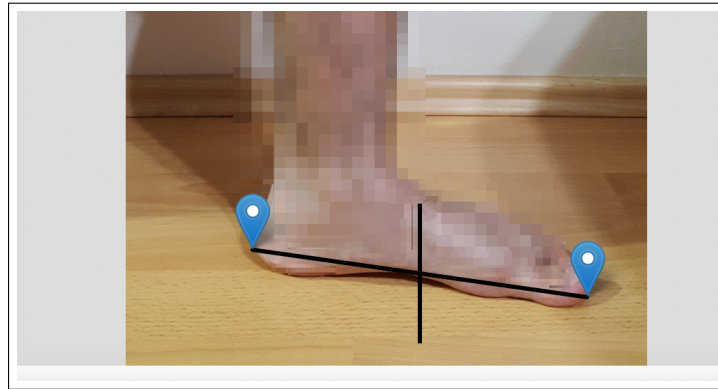


Figure 5.15. Web application critical point assistance

such as critical point assistance and information boxes. For example, the arch height index requires finding the center of the points therefore the critical point assistance displays (see Figure 5.15) the center of the points provided by the health professionals. Another example is that information boxes provide simple reminders about indexes (see Figure 5.16).

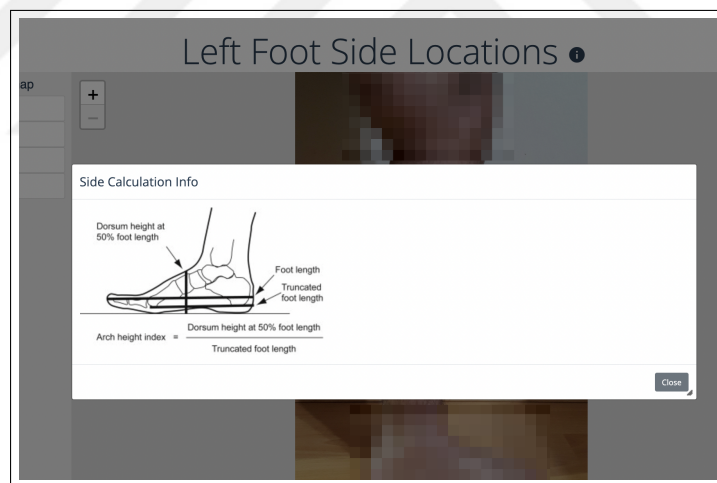


Figure 5.16. Web application information boxes

5.2.4. Services And Batch Process

The application consists of two services: web and client application service. This structure provides scalability and security. Therefore, one service deals with end-user interactions and data transmission. At the same time, web service provides healthcare professionals and batch processes. With this separation, each service is placed in different layers. Therefore, this makes it possible that only client application services have internet access, increasing

the web application security by limiting internet access. In addition, this will provide more hardware-intensive operations (image processing, deep learning) to other physical servers.

Client application services adopt the microservice architecture for scalability and respond quickly to requirements. Furthermore, to prevent unauthorized access JWT (RFC 7519) is used. Lastly, all communications have been done by HTTP protocol with an SSL (RFC 6101).

The batch process is the system's core, where the pre-diagnosis is done. The pre-diagnosis computation has three main steps: finding the region of interest, image pre-processing, and detection. Afterward, results are shared with the appropriate application and recorded in the database. In the following paragraphs, each step is detailly explained.

The initial step requires finding the foot in the image, also called the region of interest (ROI). Since all the data is collected unsupervised, the foot can be anywhere in the image. Therefore, roughly locating the foot for detail processing is essential. Deep learning algorithms are excellent in object recognition, but they require a vast amount of data to achieve this type of accuracy, so using deep learning algorithms for pre-diagnosis is close to impossible with the current dataset. As a result, the study uses pre-trained deep learning human detection algorithms to get a rough location of the foot.

The foot detection rate of the algorithms might not be as accredited as the announced results because most public datasets contain an entire human body other than a barefoot. Therefore, multiple deep learning networks such as YOLO, MIDAS, FCN Resnet 101, DeepLabV3 have been tested in small datasets for best results (see Figure 5.17).

In addition, skin detection algorithms are also tested. Initially, the YCbCr color space-based skin detection algorithm was applied, but detection results have been unsuccessful in some lighting conditions. Therefore, the zero-sum game-based skin detection algorithm proposed by Dahmani et al. [133] is also tested (see Figure 5.17). Overall, the best performance is achieved by DeepLabV3.

The second step deals with image preparation. After potential barefoot is detected, a minimum area rectangle is calculated because the foot's location and angle might not be as accurate as demonstrated in the usage document. Therefore, perspective transformation is applied



Figure 5.17. Deep learning test - DeepLab, FCN, YOLO, Skin, MIDAS (left to right)

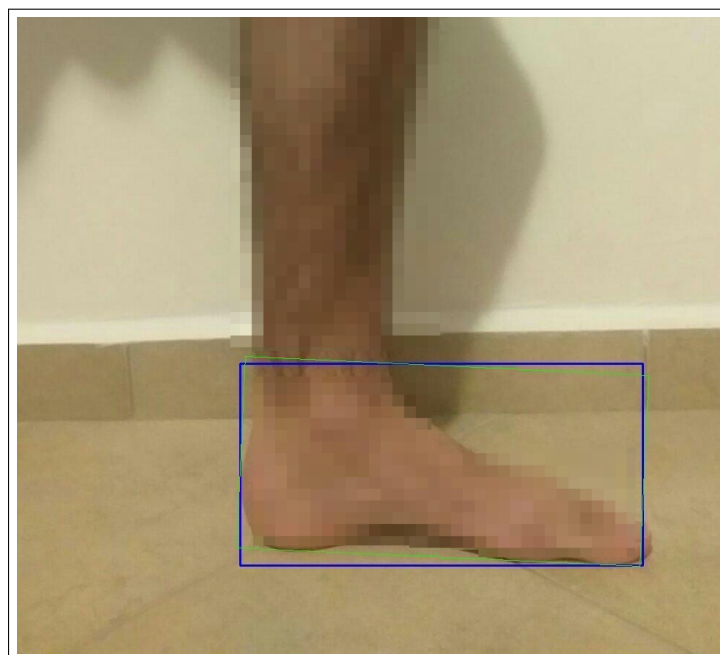


Figure 5.18. Batch process minimum bounding box (green line)

with the minimum area rectangle, eliminating potential angle problems (see Figure 5.18). In addition, this will move any foot on the image to the left corner of the image, which will ease the calculations.

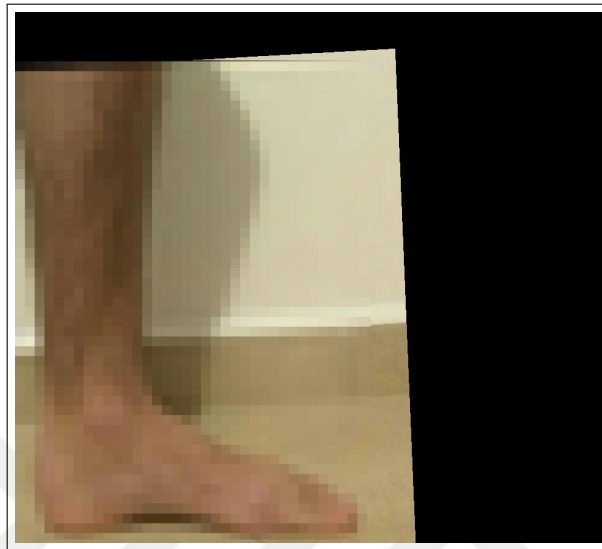


Figure 5.19. Batch process perspective transformation

After the perspective transformation (see Figure 5.19), grayscale is required for edge detection algorithms. Therefore, RGB color space is converted to HSL because HSL has a separate lightness channel that can be eliminated. After all, lightness is unrelated to foot detection. The S channel was selected for the grayscale conversion since S performed better than H in the small dataset.

The final stage includes edge detection and functionalization of the sole. The sobel algorithm is selected for edge detection with four different approaches with threshold values (see Figure 5.20);

- Applying an absolute gradient with the X direction
- Applying an absolute gradient with the Y direction
- Applying a magnitude gradient
- Applying a directional gradient with an angle threshold

After applying each configuration to the transformed image, results are merged into one with the algorithm 5.1, which reduces unwanted pixels (see Figure 5.21).

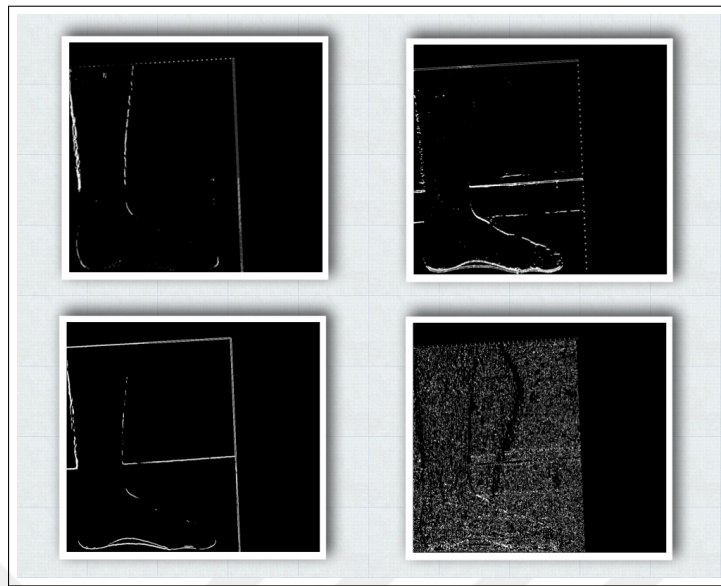


Figure 5.20. Batch Process - Sobel - Absolute Gradient X, Absolute Gradient Y, Magnitude Gradient, Directional Gradient - Left To Right

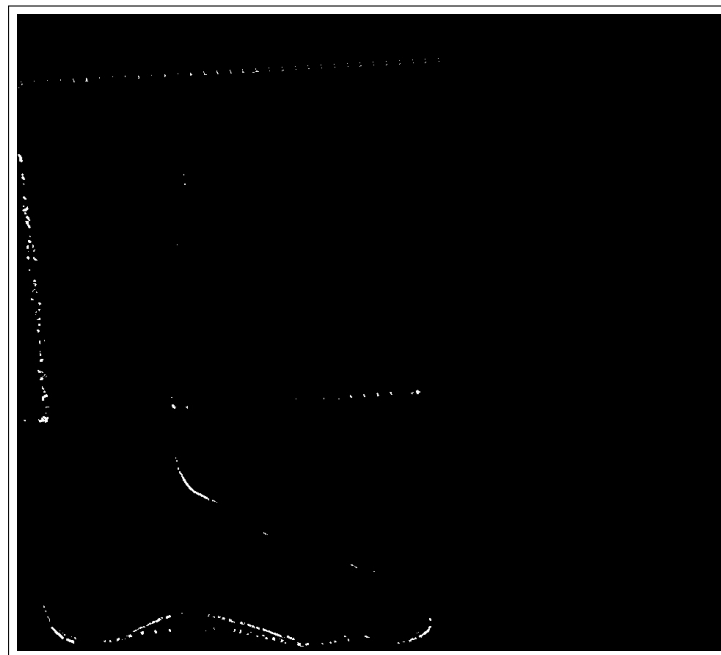


Figure 5.21. Batch process Sobel combined

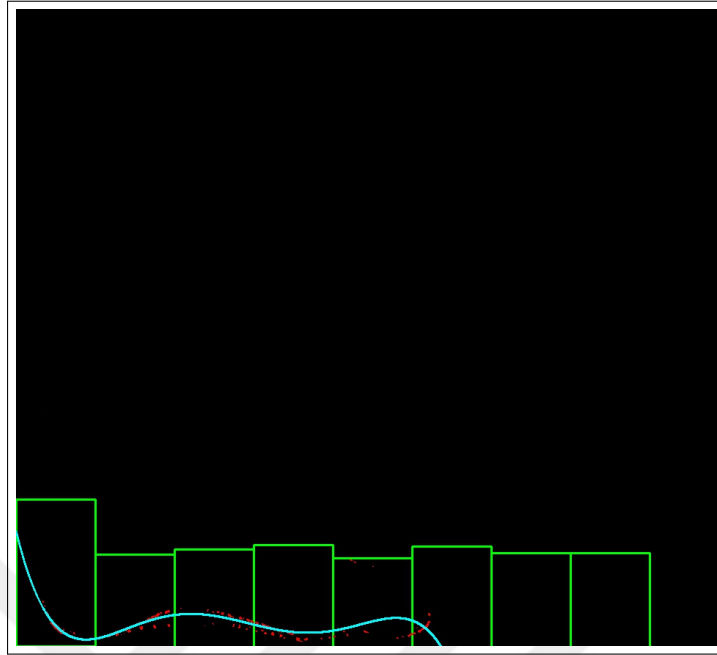


Figure 5.22. Batch process polynomial function

After the edges are extracted, polynomial regression is applied to generate the polynomial function of the solo. In addition, the fifth-order (quintic) polynomial function is selected to get matching curves of the solo. Furthermore, polynomial regression is applied with predefined windows(see Figure 5.22) because curve transitions will become smoother.

Algorithm 5.1. Merging Sobel results

```

 $ABS_X \leftarrow AbsoluteGradientX$ 
 $ABS_Y \leftarrow AbsoluteGradientY$ 
 $MAG \leftarrow MagnitudeGradient$ 
 $DIR \leftarrow DirectionalGradient$ 
for  $pixel \in image$  do
     $pixel \leftarrow (ABS_X \ \& \ ABS_Y) \text{ or } (MAG \ \& \ DIR)$ 
end for

```

Critical locations on the foot will be located based on the solo function. For example, the metatarsal point is based on the solo function's second increase. In addition, The bottom-left and bottom-right positions of the foot will be located with the solo function and minimum bounding box.

5.2.5. Database

MySQL was selected for the primary database because it provides a high degree of data integrity. In addition, the following reasons were also considered in selection: the data structure is not changing frequently, a relatively contains a small amount of data (a small percentage of users), and most importantly, data require a high degree of data integrity and security.

The database design (see Figure 5.23) was carried out within the drafts revealed due to the requirement analysis. In this process, tables were created based on the Normal forms, which were created to minimize errors and improve designs. Initially, three Normal forms were considered in the design: the first, second, and third normal forms. After that, fourth and fifth normal forms (Fifth Normal Form, 5NF) were also considered, based on the concepts of multivalued and join dependencies, respectively. Finally, all the de-normalization forms were taken into consideration during the database design.

The database was divided into eight major table groups based on usage and connections: ScanFileInfo, FootScan, File, Training, DiseaseHistory, User, Survey, and Diagnosis. Those groups help us understand the concept more clearly (see Figure 5.23).

User table group were explicitly designed for all application users, which includes end-users, healthcare professionals and admins. Its components consist of patient, foot, foot type, and gender (see Figure 5.24). In addition, users have had roles that help generalize the table, such as healthcare specialists or patients. Furthermore, foot type was added for patients with different conditions, such as a prosthetic foot.

The diagnosis table group focuses on foot diagnosis and detailed calculations methods (see Figure 5.25). The foot may have multiple diagnoses from multiple healthcare specialists or the system. In addition, diagnoses calculation details can be recorded, such as the Staheli arch index.

The ScanFileInfo, FootScan, File table groups were designed to record patient assets according to the examination type (see Figure 5.26). FootScan records examination type such as Foot Deformity or Rom Measurement. Each examination can have unlimited images

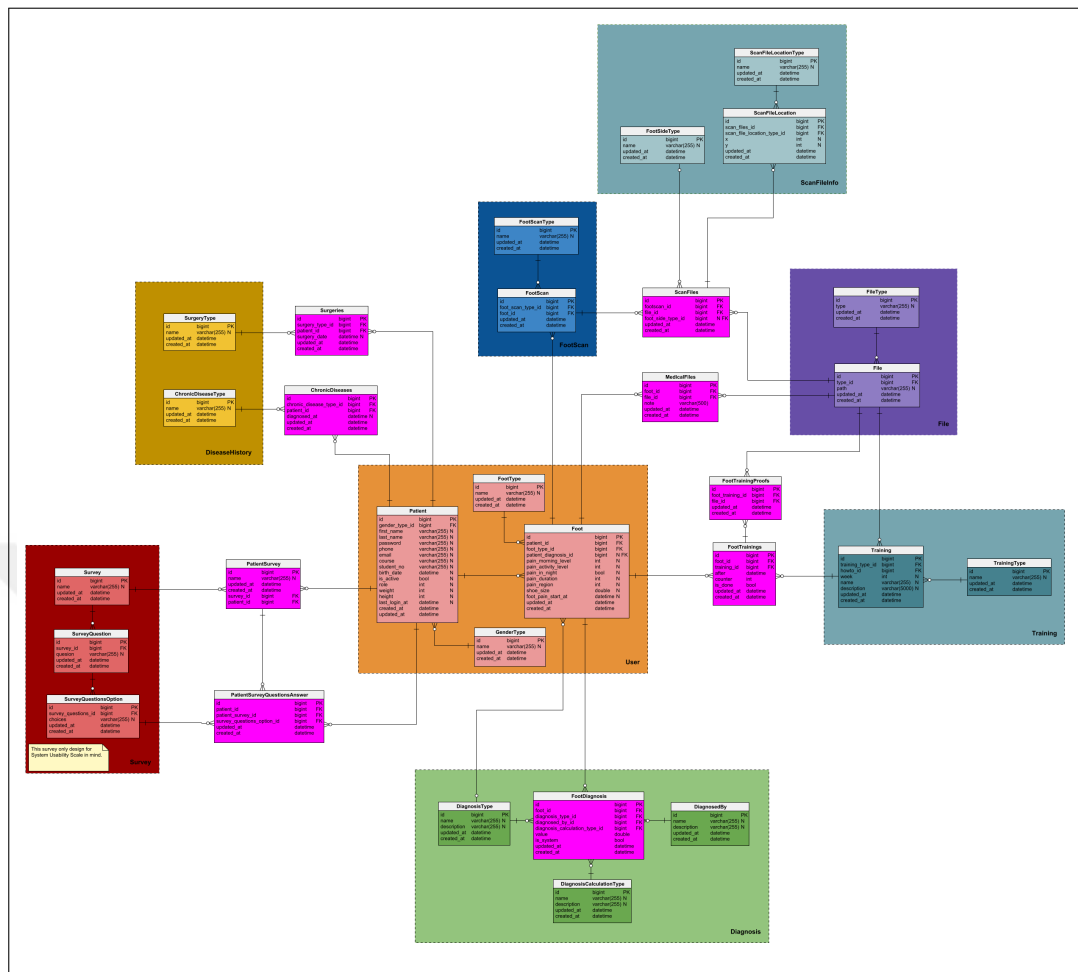


Figure 5.23. General view of database

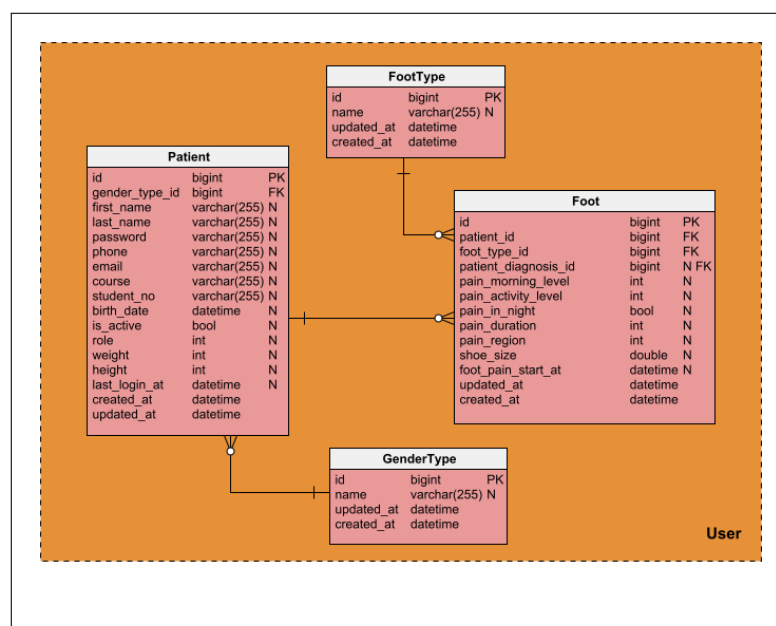


Figure 5.24. Database user group table

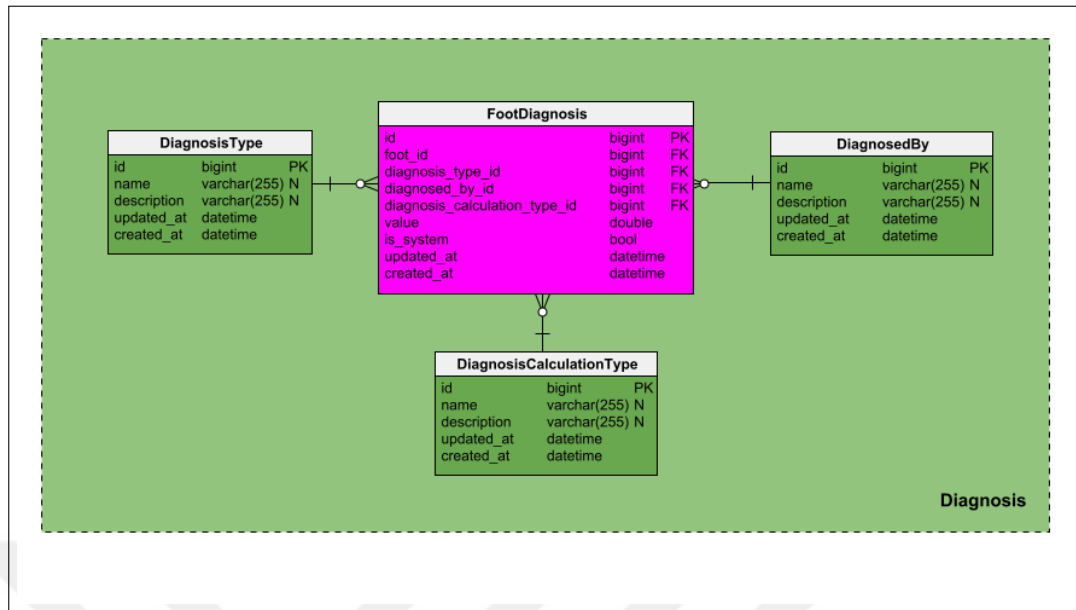


Figure 5.25. Database diagnosis group table

recorded in the ScanFiles table with a type such as Bottom, Back Side. Each image might have essential locations recorded in the ScanFileLocation table, such as the side thumb. In addition, file information such as file system locations is recorded in the File table, and each file contains a file type for processing.

Survey and Training table groups are used for the feedback and healing after deformities are detected. Those tables help the healthcare professionals track the patients' activities. In addition, the DiseaseHistory table group is used to record the previous diseases and surgeries of the patients.

5.3. TEST AND EVALUATION

In this section, the test and evaluation process of the prototype system, which was explained in detail in the previous sections, will be discussed.

In the first study, the test data were collected through mobile applications without any supervision. Therefore some of the data collected were not usable in the test and evaluation process. In addition, the data was mainly collected on the Yeditepe university students, which is a diverse population, and participants were provided with the how-to-use document.

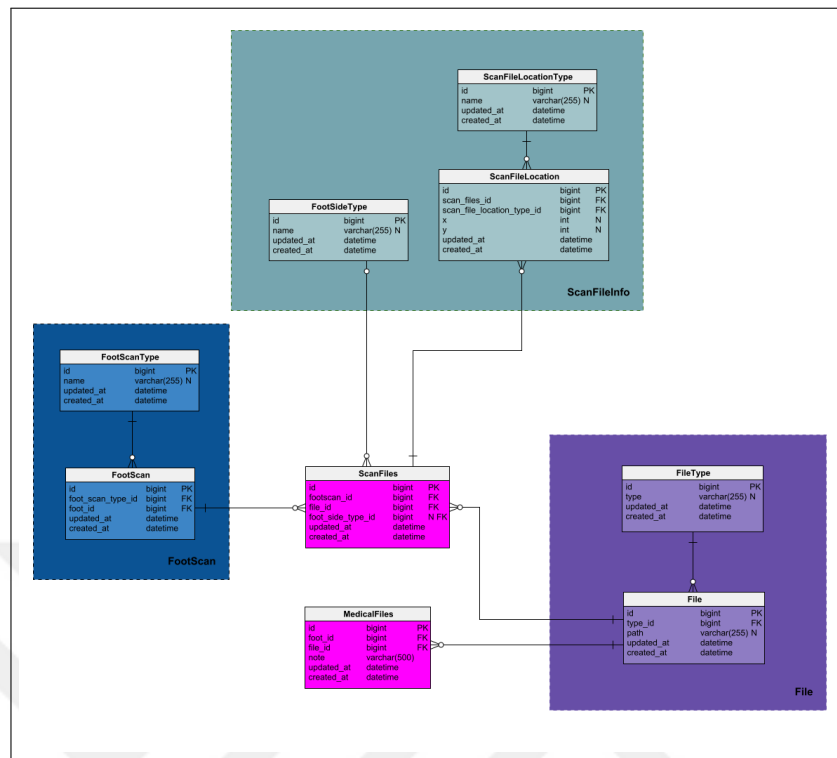


Figure 5.26. Database files group

The prototype was used by 34 participants, where 12 (35.29 percent) of them were female, and the remaining 22 (64.71 percent) were male, with an average age of 24 (see Table 5.1 for details). Data that were provided by participants were stored in an online database which is described in section 5.1.

Table 5.1. Participant statistics in study I

	Age	Weight	Height	BMI
Avarage	24.06	75.15	173.18	24.91
Max	48	120	186	37.04
Min	21	50	160	18.14

The average BMI in the population was calculated as 24.9. As seen in Figure 5.27, at least one person from each BMI category participated in the study. Therefore, this will provide sufficient data for projection.

The data collected from the users included 68 photos, 61 of them were used, but 7 of them were unusable because they were taken incorrectly. In addition, data collection was done in an unsupervised manner which caused seven unusable photos. All the data provided by the

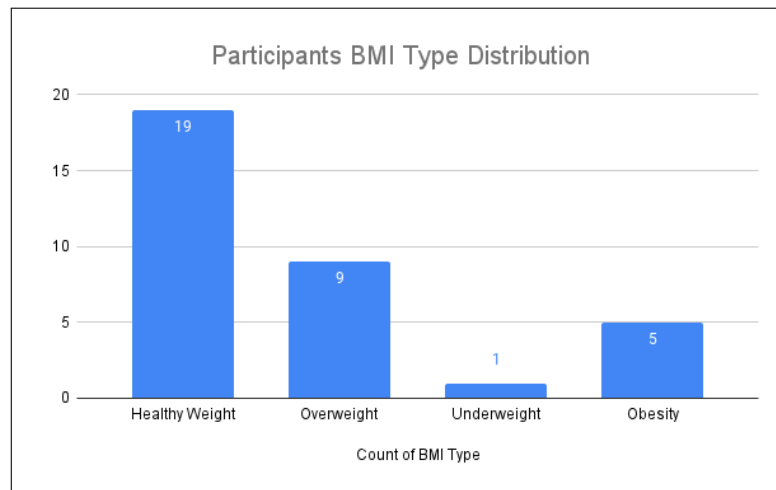


Figure 5.27. Participants BMI type distribution based on CDC in study I

participants were shared with a physician through the specialist interface, which is described in sub-section 5.2.3. The physician went through all the system data and recorded critical points to the system.

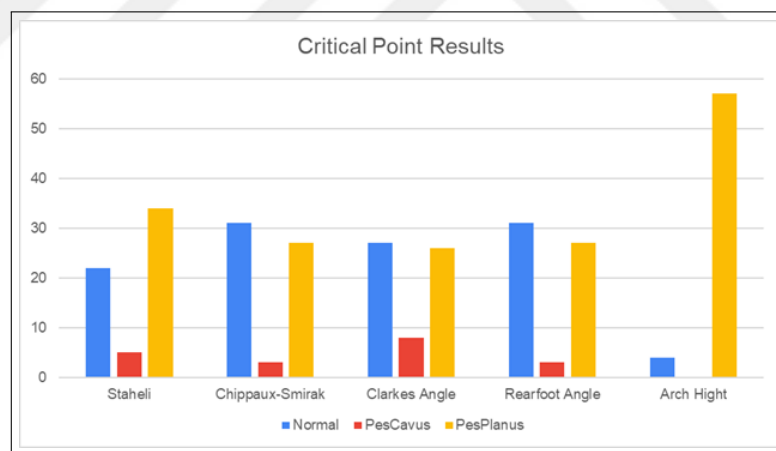


Figure 5.28. Critical point results on study I

After the physician results were collected, index calculations were conducted (see Figure 5.28). Even with specialists' critical points, the results vary based on the index type. However, even if the reference points vary, approaching specialists' results will reveal the system's usability. Therefore, the initial study has been investigating the system set critical points, and the specialist set critical point differences.

Initial results showed that the findings of the specialist and the system were 91.80 percent matched. Even though these results showed great promise, this comparison was based on

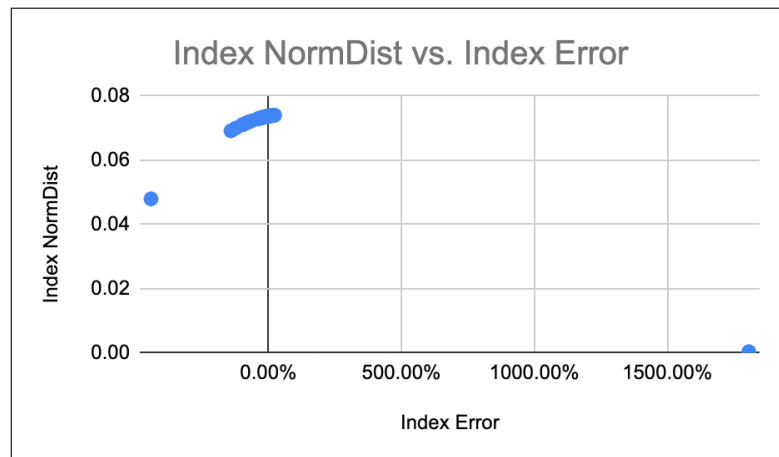


Figure 5.29. Index error normal distribution in study I

the class of the foot. Therefore, a more quantitative approach was necessary to asset the conducted test was accurate. Since automated algorithm critical points are not directly comparable with the specialist critical points (the pre-diagnosis algorithm transforms the images), point intermediate and resulting findings were compared.

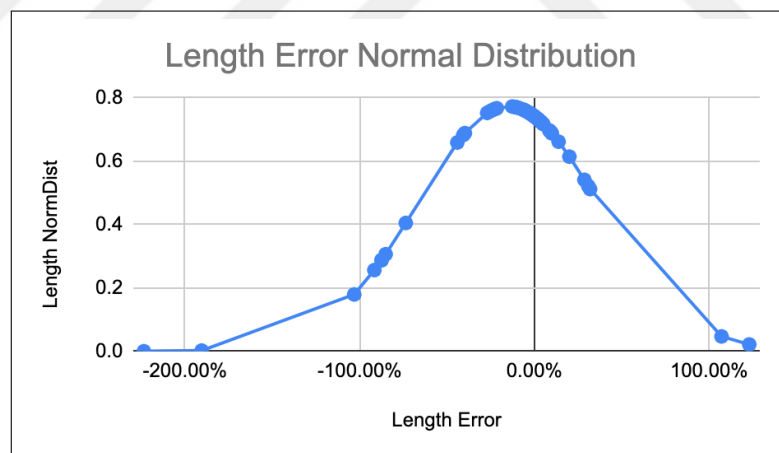


Figure 5.30. Length error normal distribution in study I

Therefore, the system index errors were calculated, and normal distribution was piloted (see Figure 5.29). The results showed that most of the normal distribution of the index error was within a small error range. On the other hand, it has been observed that there are also outliers with significant error rates. These errors were primarily due to human errors, such as misplaced critical points or photos taken from the wrong angle.

In addition, length error was also calculated to see the effect of the perspective transformation

in error calculation (see Figure 5.30). Even though those results showed similar results to index error, outliers were increased. This error increase was partially due to a perspective transformation of the images.



6. STUDY II - BACK SIDE DIAGNOSIS

In this chapter, an overview of the new pre-diagnosis (i.e., diagnosis based on back side of the foot) will be discussed—this system is based on achilles tendon. Accordingly, general system analysis and design decisions will be addressed in Section 6.1. Then, specifics of implementation will be reviewed in Section 6.2, and finally, the initial test results and evaluation phase will be discussed in Section 6.3

6.1. ANALYSIS AND DESIGN

In this section, the basic structures of the system discussed in Chapter 5 have not been changed while analyzing and designing. In addition, this chapter was profoundly focused on the small changes made to the system. Changes in requirements and design decisions will be discussed in this section.

As discussed in Section 5.3, lack of supervision causes data to be unusable and unstable test environments. Therefore, the system should be used by healthcare officials or supervised while used by participants.

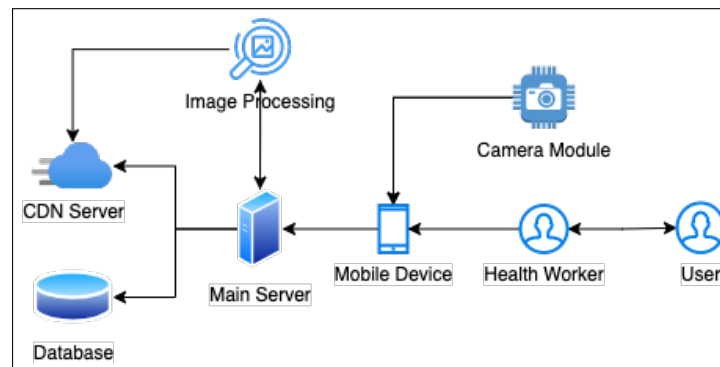


Figure 6.1. Architecture diagram of the updated system

Index results of healthcare professionals may conflict with each other due to remote image evaluations, which are discussed in Section 5.3. Therefore, healthcare professionals' diagnosis results should be added after examinations to assess the diagnosis correctly. The general structure of the converted system can be seen in Figure 6.1.

6.1.1. Prediction And Fine Tune Module

The prediction module consists of end-user interactions and pre-diagnosis. However, for improving results and getting more accurate test results, pre-diagnosis should be disabled since healthcare professionals will guide the participants.

Another module, the fine-tune module, consists of background procedures and healthcare professionals' interactions. Also, the healthcare interaction module will not be used in this module since all deformities will be diagnosed on examination, and results will be entered by mobile applications. In the following paragraphs, details of the changes will be discussed.

End-user interactions on the prediction module should be updated to be usable by the healthcare professionals for supervised data collection. The same application flow should be protected except for minor modifications. For example, healthcare professionals should be able to enter the diagnosis results of the patients into the application.

In the needs analysis, the importance of the Achilles tendon in the diagnosis process was discussed with the health professionals, and therefore it was decided to base the Achilles tendon in the automated index calculation process.

6.1.2. Index Calculation

The pre-diagnosis, the system's underlying structure, which reduces the workload of healthcare professionals, should be updated to include rearfoot angle. The rearfoot angle is a very suitable index calculation method since it is based on the Achilles tendon.

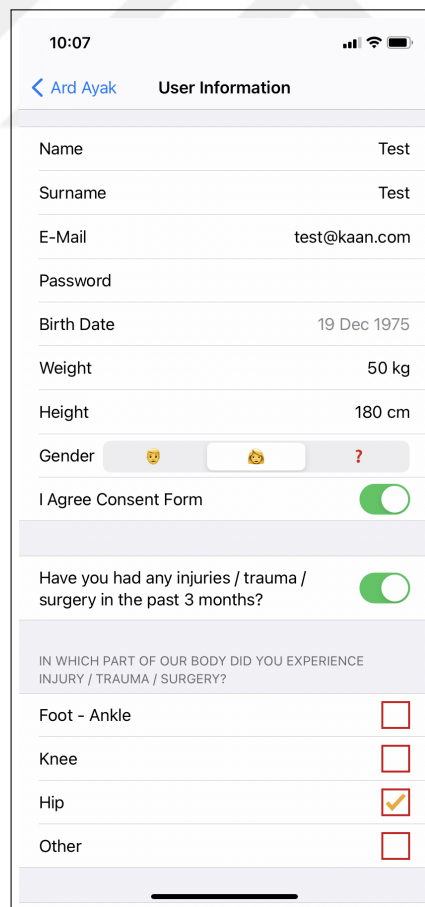
Therefore, photographs of the back of the foot should be taken, and the rearfoot angle calculations should be made. These photos must include the calf and up to the foot's heel. In addition, legs should be centered for more accurate calculations.

6.2. IMPLEMENTATION

This section discusses the implementation details and decisions of the second study. The section focuses on changes to the modules previously discussed in section 5. A more detailed explanation of changes is provided by dividing it into subsections: end-user application and services and batch process.

6.2.1. End-User Application

In this section, there was no significant change in the end-user application, only minor changes have been made to improve usability. Since users are not directly using the application, some user identification options are removed, such as phone numbers. In addition, some of the optional fields for students are also removed, such as student numbers (see Figure 6.2).



10:07

< Ard Ayak User Information

Name	Test
Surname	Test
E-Mail	test@kaan.com
Password	
Birth Date	19 Dec 1975
Weight	50 kg
Height	180 cm
Gender	☐ Male ☐ Female ☐ ?
I Agree Consent Form	☒
Have you had any injuries / trauma / surgery in the past 3 months?	
☒	
IN WHICH PART OF OUR BODY DID YOU EXPERIENCE INJURY / TRAUMA / SURGERY?	
Foot - Ankle	☐
Knee	☐
Hip	☒
Other	☐

Figure 6.2. Minor changes in end-user application in Study I

In addition to compliance with requirements analysis, some minor bugs such as default values and crashes have been fixed, and diagnostic results have been collected as if healthcare professionals had entered them, not as if they were taken from the patient.

6.2.2. Services And Batch Process

In this section, no major changes were made to the services, except for minor adjustments and bug fixes. For the different index calculations in the automated process, most of the changes were conducted in the batch process. All these changes were explained in detail below.

One of the most significant service adjustments was made in the API. Diagnosis by option was added, which were recorded in different fields in the database (see Section 5.2.5 for details). In addition, user type control was added to prevent misconfiguration.



Figure 6.3. Region of interest results in Study II

The batch process calculation was redesigned from the ground up. First, the image types

were differentiated so that the correct images were processed with the correct index types. Therefore, the foot-side type (see Section 5.2.5 for details) was used to differentiate in the scan file (i.e., in the image).

After the correct image selection, the detection of the foot and leg is very important. Therefore, deep learning algorithms are applied to find the region of interest. In this study, DeepLabV3 was used to detect the region of interest, based on the experience obtained in Section 5.2.4 (see Figure 6.3).

Following the region of interest detection, critical points were detected. There are four critical points in rearfoot angle calculation (see Section for details 4.1). Each point requires specific points to be detected in the leg. In addition, each point is very close to the midpoint of the leg (see Figure 6.4).

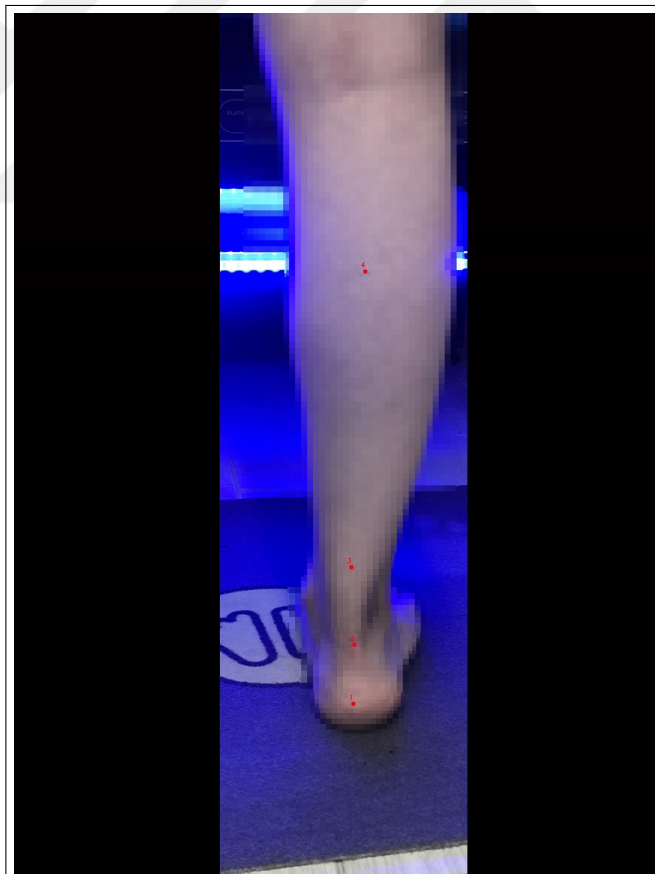


Figure 6.4. Calculated critical points for rearfoot angle (red dots)

The first point base of the calcaneus is calculated based on the region of interest. Processing is started from the bottom of the image, where the initial region is detected, wideness of the

foot area is recorded. In case of the region of interest errors, the processing is continued until 30 percent of the image width from the initial region detection. If new wideness is more significant than 50 percent of the initial detection, new area wideness is considered the calcaneus's base.

The second point determination is based on the assumption that the ankle is the thinnest area in the leg. Therefore, the narrowing of the leg is sought, following the first point discovery. After this region is found, the point that is 5 percent below the region of interest is considered as the second breakpoint. In addition, the third point had a similar discovery process. The only difference is that the third point is 5 percent up by the region of interest

The fourth point corresponds to the narrowing point of the calf. Therefore, after the third point was found, narrowing was sought in the region.

6.3. TEST AND EVALUATION

In this section, the test and evaluation process of the prototype system, which is explained in detail in the previous sections, will be discussed.

Table 6.1. Participant statistics in study II

	Age	Weight	Height	BMI
Avarage	22.67	70.89	173.11	23.24
Max	29	100	190	30.19
Min	21	50	159	19.05

In the second study, test data are collected through mobile applications under the supervision of healthcare professionals. Therefore, the collected data are evaluated before the procedure. In addition, a document describing how to use the application was given to the healthcare professionals.

The prototype was used by 9 participants, where 3 (33.33 percent) of them were female, and the remaining 6 (66,67 percent) were male, with an average age of 22.6 (see Table 6.1 for details). Data was stored in an online database which is described in section 5.1.

BMI is critical as most index calculations are based on the ratio to eliminate the effects of

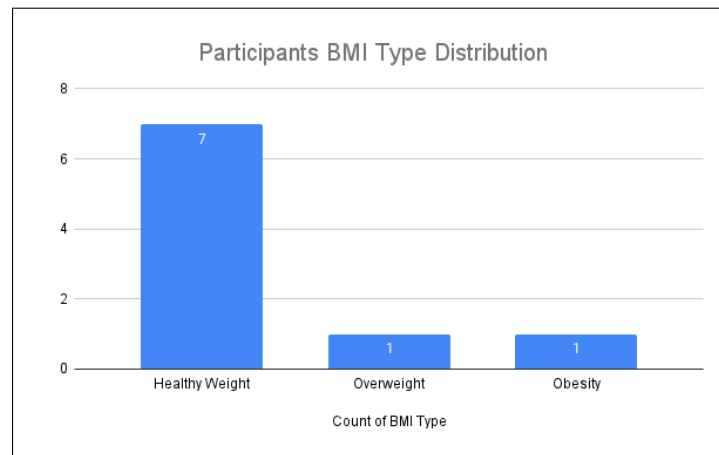


Figure 6.5. Participants BMI type distribution based on CDC in study II

weight. Population's average BMI is 23.2 (healthy weight), (see Figure 6.5). Therefore, this might not provide sufficient data for projection in different weight scales.

All 18 photos collected in the application were used because during the collection of photos all users are supervised by healthcare professionals. In addition, deformity types of the users were validated in the examination by healthcare professionals before sent to the system (see Figure 6.6).

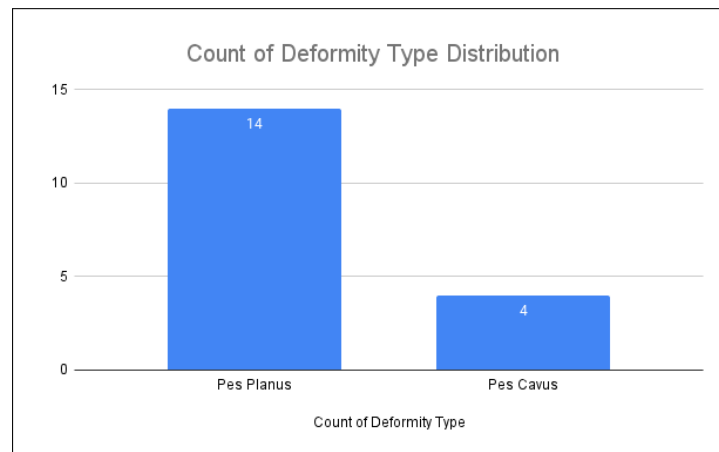


Figure 6.6. Foot deformity type distribution in study II

After the data collection, automated index calculations are conducted. Initial results showed that the findings of the healthcare officials and the system were 27,7 percent matched. This low performance may be caused by multiple factors such as image angle, color disorientation, assumptions in point location determination. However, it was thought that the biggest effect

in the evaluations was the sensitivity of the Rear foot angle to the calculation error. Because this calculation only determines the deformity according to a difference of 5 degrees. It is very difficult to detect this 5 degree difference in the photos taken via the mobile application (see Figure 6.7).

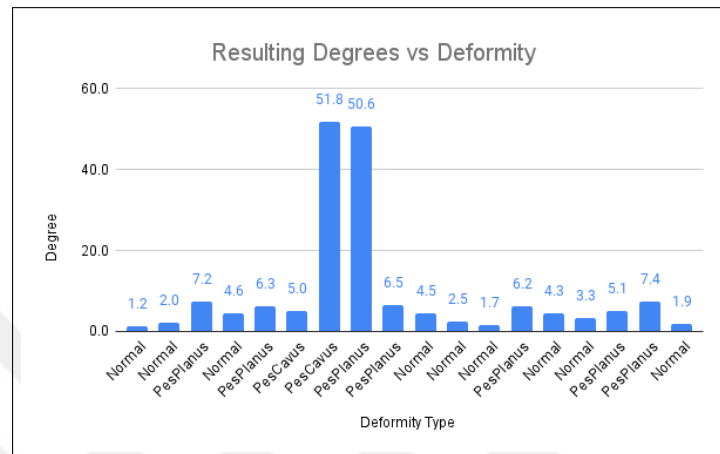


Figure 6.7. Rear foot angle vs foot deformity in automated process results

7. STUDY III - INNER SIDE DIAGNOSIS IMPROVEMENT

This chapter is focused on arch high index calculation improvements with the data provided in Chapter 6. In this study, analysis and design decisions in Section 7.1, specifics of implementation in Section 7.2, and initial test findings and evaluation phase in Section 7.3 will be discussed.

7.1. ANALYSIS AND DESIGN

Preliminary detection of potential pes planus and pes cavus patients will be provided based on the inner side photographs of the foot. This system will be based on the initial implementation of the arch height index calculation discussed in Chapter 5. In addition, this chapter profoundly focuses on improving the batch process on pre-diagnosis.

Three major fallbacks of the initial design are discovered in the test end evaluation of the process. All the fallbacks and potential solutions were discussed in the following paragraphs.

The first fallback is the differences between the line of the sole and the symmetry of the foot sole function. These differences should be reduced, and the foot function's sole should be more accurate.

The second fallback is that the bottom of the images contains a lot of noise in edge detection. By eliminating these noises, the accuracy of the foot sole function should be increased. The greedy approach could be used to improve accuracy with a relatively less computational resource.

The third fallback is that there is no error detection mechanism to reduce the error. Therefore, the function of the foot sole should not only be used for point detection but also for detecting the slope of the foot. As a result, this feature extraction should be used for detecting anomalies so that the results can be corrected.

7.2. IMPLEMENTATION

Implementation details of the changes in the batch process will be discussed separately in this section.

The first fallback is the accuracy of the foot sole function. The foot sole function calculation was designed to reduce the sudden slope changes to the function because there might be an error in edge detection. Increasing the predefined windows, where polynomial regression is applied, will improve the function's response to changes (see Figure 7.1).

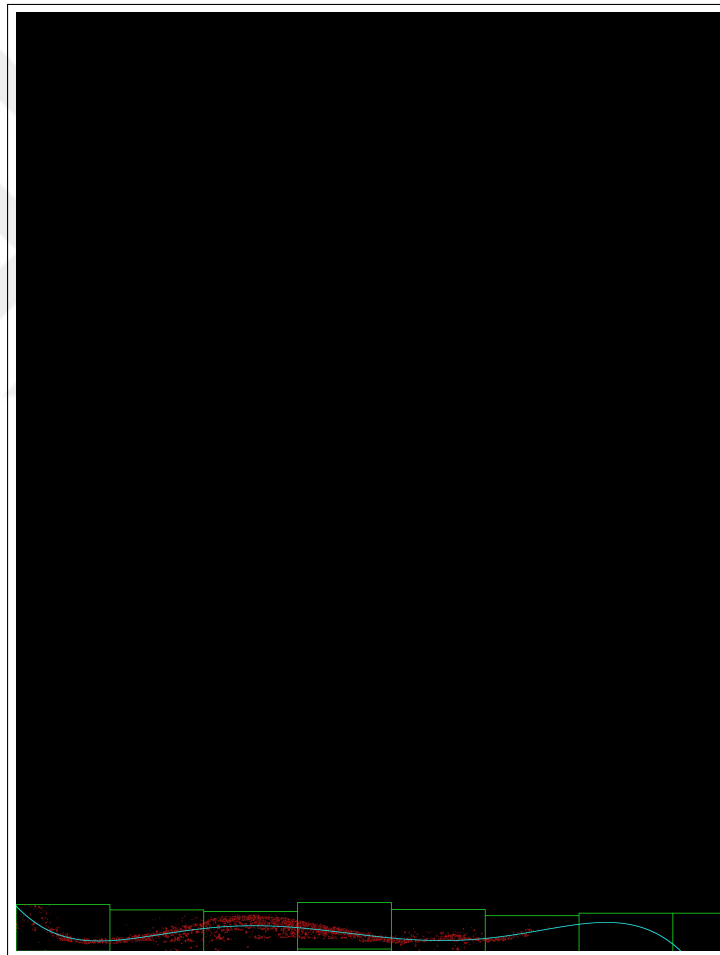


Figure 7.1. Accuracy improvement with polynomial regression window in study III

The second fallback is that the edge images contain noise. Since the bottom of the image edge points changes according to the floor decoration (carpet, faience etc.), removing the edge points of the floor will significantly improve the accuracy of the foot sole function. Since there is no edge point across the foot surface, greedy approach was used to removing

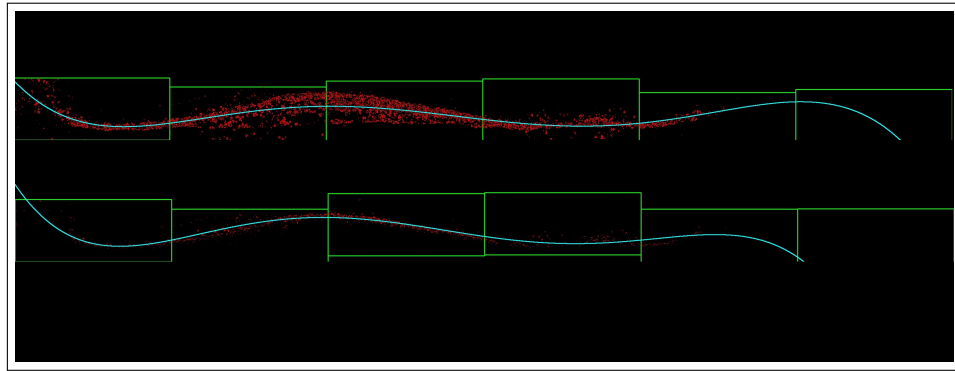


Figure 7.2. Greedy noise reduce improvement in study III

the bottom edge points (see Figure 7.2). In addition, removing 80 percent of the dots in each row was resulted in the best accuracy in the tests.

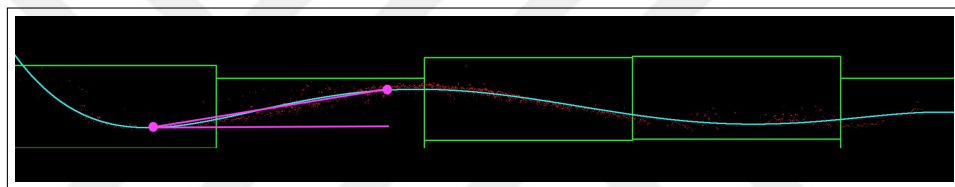


Figure 7.3. Slope calculation for error detection mechanism in study III

The final fallback is the lack of an error detection mechanism. If the slope is higher than a certain rate, it is an indication of pes cavus. Likewise, the same assumption can be made for the low slope. Therefore, with the improved foot sole function, error detection mechanism was obtained by calculating the slope of the function at certain positions. Error detection mechanism was implemented by taking two points on the foot; as the heel's peak and the foot space in the center (see Figure 7.3).

7.3. TEST AND EVALUATION

In this section, the testing and evaluation process of the prototype system, which is explained in detail in the previous sections, will be discussed. There is no data collection part in this study; the data discussed in Section 6.3 were used. Therefore, data collection and data details will be skipped.

The findings of the healthcare officials and the system were matched 83,3 percent

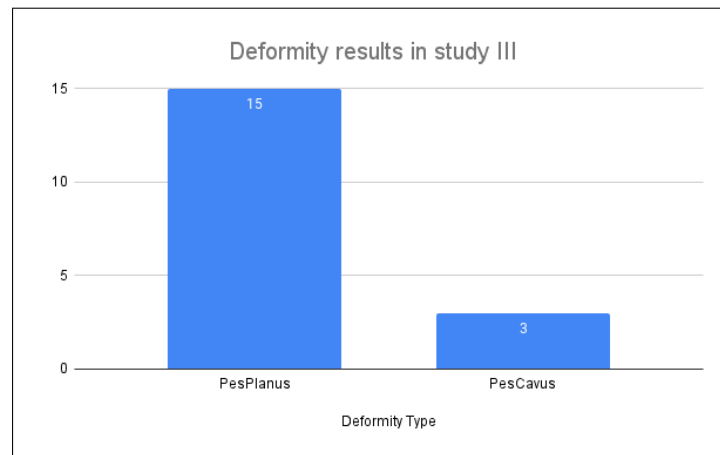


Figure 7.4. Foot deformity distribution in automated process results in study III

of the time (see Figure 7.4). One pes planus and two pes cavus mismatch in detection are recorded. Inner side diagnosis improvements were very promising (see Figure 7.5). However, the test population was not diverse. Therefore the system should be tested on a larger population to get more accurate results.

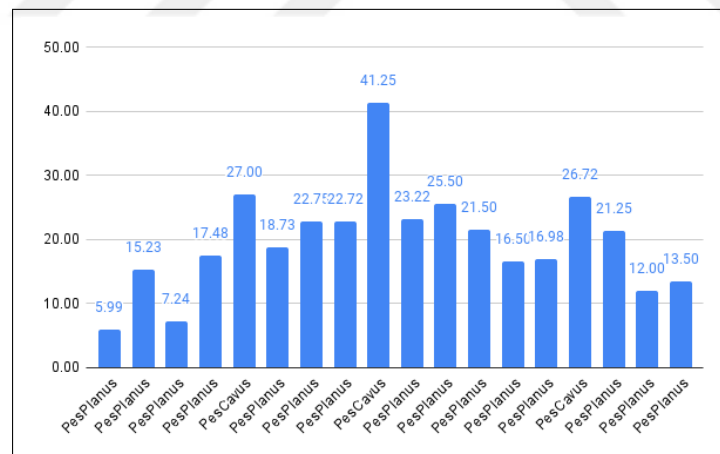


Figure 7.5. Foot deformity and calculated slope in automated process results in Study III

8. CONCLUSIONS AND FUTURE WORK

This research aims to develop a mobile pre-diagnostic platform that will assist healthcare professionals in remotely classifying and evaluating possible patients with foot deformities that is namely pes cavus and pes planus. For this purpose, three studies were conducted to accurately detect pes planus and pes cavus. The first of these studies, unlike other studies, was conducted remotely and unsupervised, so the data are not used together with other studies.

The first study focused on the general flow and requirements of the system, and the detection of foot deformation was tested by remotely receiving and processing data throughout the study. Foot deformity determination was made according to the arch height index. Study results have been verified with data provided remotely by the expert. Although the initial research results were positive and resulted in high accuracy, it cannot be said that a guaranteed conclusion was reached about the first study results since the physician made the control with the data provided by the users and did not detect the foot deformation through physical examination. This study showed that pes planus and pes cavus detection is possible using mobile applications. Therefore, second study was conducted to verify the first study's results by performing deformity detection under the supervision of healthcare professionals.

The second study focused on the acquisition of examination data and results of rearfoot angle based foot deformity detection algorithm. In this study, mobile applications were used by healthcare professionals and not by end users, to obtain supervised data. Some infrastructure changes were made to enable healthcare professionals to use the system as end-user, and as a result, more accurate data was collected. The second study shows that using a rearfoot angle based foot deformity detection algorithm was not possible for mobile based detection because the pre-diagnosis algorithm was susceptible to degree changes in the critical points lines (5 degrees). Therefore, third study was conducted to verify and improve the arch height index algorithm

The third study focused on improving foot deformity detection. Throughout this study, the data collected by the healthcare professionals in the second study and the detection method in the first study were used. In this study, the results are significantly improved compared to the second study, while reducing the shortcomings of the arch height index introduced in

the first study. The third study showed that pre-diagnose pes planus and pes cavus through a mobile phone application using image processing and deep neural networks is possible.

When the results of 3 studies are examined, the first study gives the best result with a success rate of 91.80 percent, but since the first study data were not obtained in a supervised manner, it cannot be compared with other studies. The success rates of the second and third studies are 27.7 percent and 83.3 percent, respectively. Accordingly, the results of the second study reveal the importance of selecting the correct index method for pre-diagnosis.

Although test trials were conducted in various populations, tests were carried out with limited participants, especially in the second and third studies. Therefore, the results in larger populations may be different.

Many different adaptations, experiments and studies could be done in the future. These are, first, that better results can be achieved by collecting more data and moving manual feature extraction to deep learning exercises, and secondly, problems in data collection can be eliminated by placing various controls in end-user applications and finally, the development of detection algorithms based on other pes planus and pes cavus diagnostic methods accepted in the literature.

REFERENCES

1. Russell S, Norvig P. *Artificial Intelligence: A Modern Approach*. 3rd ed. Prentice Hall; 2010.
2. Fortmann-Roe S. Bias and variance [cited 2021 24 December]. Available from: <http://scott.fortmann-roe.com/docs/BiasVariance.html>.
3. Kurihara T, Takaki Y. Shading of a computer-generated hologram by zone plate modulation. *Optics express*. 2012;20(4):3529-40.
4. Franson J. Electromagnetic Spectrum [cited 2021 24 December]. Available from: <https://sites.google.com/a/coe.edu/principles-of-structural-chemistry/relationship-between-light-and-matter/electromagnetic-spectrum>.
5. Kim J, Kim JB, Lee WC. Dynamic medial column stabilization using flexor hallucis longus tendon transfer in the surgical reconstruction of flatfoot deformity in adults. *Foot and Ankle Surgery*. 2021;27(8):920-7.
6. Bras S. Pes Cavus. Physiopedia. 2020. Available from: https://www.physio-pedia.com/Pes_cavus.
7. Heravi H, Ebrahimi A, Nikzad S, Olyae E, Zamani Y. Low Price Foot Pressure Distribution Screening Technique: Optical Podoscope with Accurate Foot Print Segmentation using Hidden Markov Random Field Model. *Journal of Biomedical Physics and Engineering*. 2018 01;10.
8. Deniz G, Kavaklı A, Ögetürk M, Öztürk D, Tatar N, Perilioğlu AZ. Çocuklardaki Fleksibl Pes Planusun Yüklü ve Yüksüz Radyografilerle Değerlendirilmesi. *Fırat Üniversitesi Sağlık Bilimleri Tıp Dergisi*. 2014;28(3):129-32.
9. Davies AM, Whitehouse RW, Jenkins JP. *Imaging of the foot & ankle: techniques and applications*. Springer Science & Business Media; 2012.
10. Lorenzzoni DP. Meary's angle. 2020. Available from: <https://radiopaedia.org/cases/mearys-angle>.

11. Ozer CM, Barut C. Evaluation of the sole morphology of professional football players. *International SportMed Journal*. 2012;13(1):8-17.
12. Miller EE, Whitcome KK, Lieberman DE, Norton HL, Dyer RE. The effect of minimal shoes on arch structure and intrinsic foot muscle strength. *Journal of Sport and Health Science*. 2014;3(2):74-85.
13. Langley B, Cramp M, Morrison SC. Clinical measures of static foot posture do not agree. *Journal of foot and ankle research*. 2016;9(1):1-6.
14. Veronez MR, Florêncio de Souza S, Matsuoka MT, Reinhardt A, Macedônio da Silva R. Regional mapping of the geoid using GNSS (GPS) measurements and an artificial neural network. *Remote sensing*. 2011;3(4):668-83.
15. Bre F, Gimenez JM, Fachinotti VD. Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings*. 2018;158:1429-41.
16. Quiza R, Davim J. Computational modeling of machining systems. *Intelligent machining: modeling and optimization of the machining processes and systems London: ISTE*. 2009:173-213.
17. Ferracuti N, Norscini C, Frontoni E, Gabellini P, Paolanti M, Placidi V. A business application of RTLS technology in Intelligent Retail Environment: Defining the shopper's preferred path and its segmentation. *Journal of Retailing and Consumer Services*. 2019;47:184-94.
18. Veli L, Aydınok H. Shoes and foot health Shoes and foot health. *Acta Orthopaedica et Traumatologica Turcica*. 2004;24(3):188-94.
19. Çilli F, Pehlivan Ö, Keklikçi K, Mahiroğullari M, Kuşkucu M. Prevalence of flatfoot in Turkish male adolescents. *Eklemler hastalıkları ve cerrahisi= Joint diseases & related surgery*. 2009;20(2):90-2.
20. Pfeiffer M, Kotz R, Ledl T, Hauser G, Sluga M. Prevalence of flat foot in preschool-aged children. *Pediatrics*. 2006;118(2):634-9.
21. Abdel Fattah M, Hassanin M, Felembane F, Nassaane MT. Flat foot among Saudi

- Arabian army recruits: prevalence and risk factors. *EMHJ-Eastern Mediterranean Health Journal*, 12 (1-2), 211-217, 2006. 2006.
22. Chen JP, Chung MJ, Wang MJ. Flatfoot prevalence and foot dimensions of 5-to 13-year-old children in Taiwan. *Foot & ankle international*. 2009;30(4):326-32.
 23. Kharbuja R, Dhungel S. Prevalence Of Pes Cavus And Pes Planus Among School Going Children Of Bhaktapur District, Nepal. *Nepal Med Coll J*. 2017;19(1):9-12.
 24. Turing AM. Computing Machinery and Intelligence. *Mind*. 1950 January 01;49:433-60.
 25. Winograd T. Understanding natural language. *Cognitive Psychology*. 1972;3(1):1 191.
 26. Mauldin ML. CHATTERBOTS, TINYMUDS, and the Turing Test: Entering the Loebner Prize Competition. *AAAI*. AAAI Press / The MIT Press. 1994:16-21.
 27. Preston J, Bishop M. *Views into the Chinese room: New essays on Searle and artificial intelligence*. Oxford University Press; 2002.
 28. Bringsjord S, Schimanski B. What is Artificial Intelligence? Psychometric AI as an Answer. *IJCAI*. Morgan Kaufmann. 2003:887-93.
 29. Bostrom N. Ethical issues in advanced artificial intelligence. *Science fiction and philosophy: from time travel to superintelligence*. 2003:277-84.
 30. Davis R, Lenat DB. *Knowledge-Based Systems in Artificial Intelligence: 2 Case Studies*. McGraw-Hill, Inc.; 1982.
 31. Dale R, Moisl H, Somers H. *Handbook of natural language processing*. CRC press; 2000.
 32. Zadeh L. Fuzzy logic= computing with words. *IEEE Transactions on Fuzzy Systems*. 1996;4(2):103-11.
 33. Brady M. Artificial intelligence and robotics. *Robotics and Artificial Intelligence*. Springer. 1984:47-63.
 34. Michie D, Spiegelhalter DJ, Taylor CC. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood; 1994.

35. Sathya R, Abraham A, et al. Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*. 2013;2(2):34-8.
36. Geman S, Bienenstock E, Doursat R. Neural networks and the bias/variance dilemma. *Neural computation*. 1992;4(1):1-58.
37. James GM. Variance and bias for general loss functions. *Machine learning*. 2003;51(2):115-35.
38. Tan PN, Steinbach M, Kumar V. *Introduction to data mining*. Pearson Education India; 2016.
39. Harrington P. *Machine learning in action*. Simon and Schuster; 2012.
40. Cover T, Hart P. Nearest neighbor pattern classification. *IEEE transactions on information theory*. 1967;13(1):21-7.
41. Hall P, Park BU, Samworth RJ. Choice of neighbor order in nearest-neighbor classification. *the Annals of Statistics*. 2008;36(5):2135-52.
42. Hssina B, Merbouha A, Ezzikouri H, Erritali M. A comparative study of decision tree ID3 and C4. 5. *International Journal of Advanced Computer Science and Applications*. 2014;4(2):13-9.
43. Jin C, De-Lin L, Fen-Xiang M. An improved ID3 decision tree algorithm. *2009 4th International Conference on Computer Science & Education*. IEEE. 2009:127-30.
44. Chao WL. Machine learning tutorial. *Digital Image and Signal Processing*. 2011.
45. Sonawale SA, Ade R. Dimensionality reduction: an effective technique for feature selection. *International Journal of Computer Applications*. 2015;117(3).
46. Biricik G. Comparing the impacts of dimension reduction methods that use class labels on text classification. *2012 20th Signal Processing and Communications Applications Conference (SIU)*. IEEE. 2012:1-4.
47. Schölkopf B, Smola A, Müller KR. Kernel principal component analysis. *International*

- conference on artificial neural networks*. Springer. 1997:583-8.
48. Sutton RS, Barto AG. *Reinforcement learning: An introduction*. MIT press; 2018.
 49. Alpaydin E. *Introduction to machine learning*. MIT press; 2020.
 50. Tijssma AD, Drugan MM, Wiering MA. Comparing exploration strategies for q-learning in random stochastic mazes. *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2016:1-8.
 51. Guo M, Liu Y, Malec J. A new Q-learning algorithm based on the metropolis criterion. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*. 2004;34(5):2140-3.
 52. Whitehead SD. Complexity and Cooperation in Q-Learning. *ML*. 1991:.
 53. Watkins CJCH. Learning from delayed rewards. 1989.
 54. Ha D, Schmidhuber J. World models. *arXiv preprint arXiv:180310122*. 2018.
 55. Racanière S, Weber T, Reichert DP, Buesing L, Guez A, Rezende D, et al. Imagination-augmented agents for deep reinforcement learning. *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017:5694-705.
 56. Bansal S, Calandra R, Chua K, Levine S, Tomlin C. Mbmf: Model-based priors for model-free reinforcement learning. *arXiv preprint arXiv:170903153*. 2017.
 57. Deisenroth M, Rasmussen CE. PILCO: A model-based and data-efficient approach to policy search. *Proceedings of the 28th International Conference on machine learning (ICML-11)*. Citeseer. 2011:465-72.
 58. Pitas I. *Digital image processing algorithms and applications*. John Wiley & Sons; 2000.
 59. Gonzalez RC, Woods RE. *Digital image processing*. Pearson; 2018.
 60. Umbaugh SE. *Computer imaging: digital image analysis and processing*. CRC press; 2005.

61. Costa LdFD, Cesar Jr RM. *Shape analysis and classification: theory and practice*. CRC Press, Inc.; 2000.
62. Russ JC. *The image processing handbook*. CRC press; 2010.
63. Getreuer P. A survey of Gaussian convolution algorithms. *Image Processing On Line*. 2013;2013:286-310.
64. Prewitt JM. Object enhancement and extraction. *Picture processing and Psychopictorics*. 1970;10(1):15-9.
65. Bovik AC. *Handbook of image and video processing*. Academic press; 2010.
66. Moeslund TB. *Image and video processing*. Institut for Medieteknologi, Aalborg Universitet; 2009.
67. Woźniacka R, Bac A, Matusik S, Szczygiał E, Ciszek E. Body weight and the medial longitudinal foot arch: high-arched foot, a hidden problem? *European journal of pediatrics*. 2013;172(5):683-91.
68. Kohls-Gatzoulis J, Woods B, Angel J, Singh D. The prevalence of symptomatic posterior tibialis tendon dysfunction in women over the age of 40 in England. *Foot and Ankle Surgery*. 2009;15(2):75-81.
69. DA B, DR S. " IDIOPATHIC" PES CAVUS: AN INVESTIGATION INTO ITS AETIOLOGY. *British Medical Journal*. 1963;2(5358):659-61.
70. Gün K, Saridoğan M, Uysal Ö. Pes Planus Tanısında Ayak İzi ve Radyografik Ölçüm Yöntemlerinin Korelasyonu. *Turkish Journal of Physical Medicine & Rehabilitation/Turkiye Fiziksel Tip ve Rehabilitasyon Dergisi*. 2012;58(4).
71. Mickle KJ, Steele JR, Munro B. The Feet of Overweight and Obese Young Children: Are They Flat or Fat? *Obesity*. 2006;14.
72. Woźniacka R, Bac A, Matusik S, Szczygiał E, Ciszek E. Body weight and the medial longitudinal foot arch: high-arched foot, a hidden problem? *European Journal of Pediatrics*. 2013;172:683 691.

73. Kohls-Gatzoulis JA, Woods BJ, Angel JC, Singh D. The prevalence of symptomatic posterior tibialis tendon dysfunction in women over the age of 40 in England. *Foot and ankle surgery : official journal of the European Society of Foot and Ankle Surgeons*. 2009;15 2:75-81.
74. Brewerton D, Sandifer PH, Sweetnam DR. "IDIOPATHIC" PES CAVUS: AN INVESTIGATION INTO ITS AETIOLOGY. *British medical journal*. 1963;2 5358:659-61.
75. Yalçın N, Esen E, Kanatlı U, Yetkin H. Medial longitudinal arkin değerlendirilmesi: dinamik plantar basınç ölçüm sistemi ile radyografik yöntemlerin karşılaştırılması. *Acta Orthop Traumatol Turc*. 2010;44(3):241-5.
76. Smith DG, Barnes BC, Sands AK, Boyko EJ, Ahroni JH. Prevalence of Radiographic Foot Abnormalities in Patients with Diabetes. *Foot & Ankle International*. 1997;18:342 346.
77. Winfeld MJ, Winfeld BE. Management of pediatric foot deformities: an imaging review. *Pediatric Radiology*. 2019;49:1678 1690.
78. Chen KC, Yeh CJ, Kuo J, Hsieh CL, Yang SF, Wang CH. Footprint analysis of flatfoot in preschool-aged children. *European Journal of Pediatrics*. 2010;170:611-7.
79. Pauk J, Ihnatouski M, Najafi B. Assessing plantar pressure distribution in children with flatfoot arch: application of the Clarke angle. *Journal of the American Podiatric Medical Association*. 2014;104 6:622-32.
80. Kanatlı U, Yetkin H, Cila E. Footprint and Radiographic Analysis of the Feet. *Journal of Pediatric Orthopaedics*. 2001;21:225-8.
81. Yalçın N, Esen E, Kanatlı U, Yetkin H. Evaluation of the medial longitudinal arch: a comparison between the dynamic plantar pressure measurement system and radiographic analysis. *Acta orthopaedica et traumatologica turcica*. 2010;44 3:241-5.
82. Menz HB, Munteanu SE. Validity of 3 clinical techniques for the measurement of static foot posture in older people. *Journal of Orthopaedic & Sports Physical Therapy*.

2005;35(8):479-86.

83. Didia BC, Omu ET, Obuoforibo AA. The Use of Footprint Contact Index II for Classification of Flat Feet in a Nigerian Population. *Foot & Ankle International*. 1987;7:285-289.
84. Cavanagh PR, Rodgers MM. The arch index: a useful measure from footprints. *Journal of biomechanics*. 1987;20 5:547-51.
85. Igbigbi PS, Msamati BC. The footprint ratio as a predictor of pes planus: a study of indigenous Malawians. *The Journal of foot and ankle surgery : official publication of the American College of Foot and Ankle Surgeons*. 2002;41 6:394-7.
86. Gun K, Saridogan M, Uysal O. The correlation between footprint and radiographic measurements in flatfoot. 2012.
87. Buldt AK, Levinger P, Murley GS, Menz HB, Nester CJ, Landorf KB. Foot posture is associated with kinematics of the foot during gait: A comparison of normal, planus and cavus feet. *Gait & posture*. 2015;42 1:42-8.
88. Gait Analysis Product. 2021. Available from: <https://www.sennotech.com/en/index.php>.
89. Gait Analysis Product. 2021. Available from: https://alfoots.com:5000/en/sub/02_sub/02_sub02.php.
90. Motion capture cameras: The full range from Vicon. 2021. Available from: <https://www.vicon.com/hardware/cameras/>.
91. Buldt AK, Forghany S, Landorf KB, Levinger P, Murley GS, Menz HB. Foot posture is associated with plantar pressure during gait: A comparison of normal, planus and cavus feet. *Gait & posture*. 2018;62:235-40.
92. Keukenkamp R, Busch-Westbroek T, Barn R, Woodburn J, Bus S. Foot ulcer recurrence, plantar pressure and footwear adherence in people with diabetes and Charcot midfoot deformity: A cohort analysis. *Diabetic Medicine*. 2020 11;38.
93. Bosch K, Gerß J, Rosenbaum D. Development of healthy children's feet—Nine-year

- results of a longitudinal investigation of plantar loading patterns. *Gait & Posture*. 2010;32(4):564-71. Available from: <https://www.sciencedirect.com/science/article/pii/S0966636210002274>.
94. emed Pressure Detection Device. 2021. Available from: <https://www.novel.de/products/emed/>.
 95. emed Pressure Detection Device. 2021. Available from: <https://medilogic.com/en/pressure-measuring-platform-nx/>.
 96. Flores DV, Mejía Gómez C, Fernández Hernando M, Davis MA, Pathria MN. Adult acquired flatfoot deformity: anatomy, biomechanics, staging, and imaging findings. *Radiographics*. 2019;39(5):1437-60.
 97. Yates B, Merriman LM. *Merriman's assessment of the lower limb*. Elsevier Health Sciences; 2009.
 98. Vanderwilde R, Staheli L, Chew D, Malagon V. Measurements on radiographs of the foot in normal infants and children. *The Journal of bone and joint surgery American volume*. 1988;70(3):407-15.
 99. Banks AS. *McGlamry's comprehensive textbook of foot and ankle surgery*. vol. 1. Lippincott Williams & Wilkins; 2001.
 100. Almaawi A, Alotaibi N, Alsubaie M, Altwaijri N, Alduraibi K, Awwad W, et al. Flatfoot Prevalence in Riyadh City Saudi Arabia And Its Association with Obesity, Using Three Footprint Indices; Clark's Angle, Chippaux-Smirak Index, and Staheli Index. *Orthopedics and Rheumatology Open Access Journals*. 2019;15(2):52-8.
 101. Hegazy F, Aboelnasr E, Abuzaid M, Kim IJ, Salem Y. Comparing Validity and Diagnostic Accuracy of Clarke's Angle and Foot Posture Index-6 to Determine Flexible Flatfoot in Adolescents: A Cross-Sectional Investigation. *Journal of Multidisciplinary Healthcare*. 2021;14:2705.
 102. Igbigbi PS, Msamati BC, Shariff MB. Arch index as a predictor of pes planus: a comparative study of indigenous Kenyans and Tanzanians. *Journal of the American*

- Podiatric Medical Association*. 2005;95(3):273-6.
103. Hillstrom HJ, Song J, Kraszewski AP, Hafer JF, Mootanah R, Dufour AB, et al. Foot type biomechanics part 1: structure and function of the asymptomatic foot. *Gait & posture*. 2013;37(3):445-51.
 104. Huerta JP, García JMA, Matamoros EC, Matamoros JC, Martínez TD. Relationship of body mass index, ankle dorsiflexion, and foot pronation on plantar fascia thickness in healthy, asymptomatic subjects. *Journal of the American Podiatric Medical Association*. 2008;98(5):379-85.
 105. Jonson LSR, Gross MT. Intraexaminer reliability, interexaminer reliability, and mean values for nine lower extremity skeletal measures in healthy naval midshipmen. *Journal of Orthopaedic & Sports Physical Therapy*. 1997;25(4):253-63.
 106. Goodfellow I, Bengio Y, Courville A. *Deep learning*. MIT press; 2016.
 107. LeCun Y, Bengio Y, Hinton G. Deep learning. *nature*. 2015;521(7553):436-44.
 108. Chen T, Goodfellow I, Shlens J. Net2net: Accelerating learning via knowledge transfer. *arXiv preprint arXiv:151105641*. 2015.
 109. Huang JT, Li J, Yu D, Deng L, Gong Y. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2013:7304-8.
 110. Baldi P, Chauvin Y. Neural networks for fingerprint recognition. *neural computation*. 1993;5(3):402-18.
 111. Tian Y, Pei K, Jana S, Ray B. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. *Proceedings of the 40th international conference on software engineering*. 2018:303-14.
 112. Melin P, Urias J, Solano D, Soto M, Lopez M, Castillo O. Voice Recognition with Neural Networks, Type-2 Fuzzy Logic and Genetic Algorithms. *Engineering Letters*. 2006;13(3).
 113. Hsieh WW, Tang B. Applying neural network models to prediction and data analysis

- in meteorology and oceanography. *Bulletin of the American Meteorological Society*. 1998;79(9):1855-70.
114. Oh IS, Suen CY. A class-modular feedforward neural network for handwriting recognition. *pattern recognition*. 2002;35(1):229-44.
 115. Medsker L, Jain LC. *Recurrent neural networks: design and applications*. CRC press; 1999.
 116. Mikolov T, Kombrink S, Burget L, Černocký J, Khudanpur S. Extensions of recurrent neural network language model. *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2011:5528-31.
 117. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*. 2014.
 118. Miao Y, Gowayyed M, Metze F. EESSEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE. 2015:167-74.
 119. Mao J, Xu W, Yang Y, Wang J, Huang Z, Yuille A. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*. 2014.
 120. Garg V, Markhedkar VA, Lale SS, Raghunandan T. Video Tagging and Recommender System Using Deep Learning. *Innovations in Computational Intelligence and Computer Vision*. Springer. 2021:302-10.
 121. Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, et al. Recent advances in convolutional neural networks. *Pattern Recognition*. 2018;77:354-77.
 122. Bouvrie J. Notes on convolutional neural networks. 2006.
 123. Lavin A, Gray S. Fast algorithms for convolutional neural networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016:4013-21.
 124. Nogueira V, Oliveira H, Silva JA, Vieira T, Oliveira K. RetailNet: A deep learning approach for people counting and hot spots detection in retail stores. *2019*

- 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE. 2019:155-62.
125. Rausch V, Hansen A, Solowjow E, Liu C, Kreuzer E, Hedrick JK. Learning a deep neural net policy for end-to-end control of autonomous vehicles. *2017 American Control Conference (ACC)*. IEEE. 2017:4914-9.
 126. Yang S, Luo P, Loy CC, Tang X. From facial parts responses to face detection: A deep learning approach. *Proceedings of the IEEE international conference on computer vision*. 2015:3676-84.
 127. Girshick R. Fast r-cnn. *Proceedings of the IEEE international conference on computer vision*. 2015:1440-8.
 128. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*. 2016;39(6):1137-49.
 129. Redmon J, Farhadi A. Yolov3: An incremental improvement. *arXiv preprint arXiv:180402767*. 2018.
 130. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015:3431-40.
 131. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016:770-8.
 132. Chen LC, Papandreou G, Schroff F, Adam H. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:170605587*. 2017.
 133. Dahmani D, Cheref M, Larabi S. Zero-sum game theory model for segmenting skin regions. *Image and Vision Computing*. 2020;99:103925.