SKIN DISEASE PREDICTION USING DEEP NEURAL NETWORK

by

Ahmet Görünücü
Emre Dindar
Zafer Emre Ocak

CSE497 / CSE498 Engineering Project report submitted to Faculty of Engineering in partial fulfillment of the requirements for the degree of

BACHELOR OF SCIENCE

Supervised by:
Assoc. Prof. Ali Fuat Alkaya

Marmara University, Faculty of Engineering

Computer Engineering Department

2019

ABSTRACT

In today's world, one of the deadliest diseases are cancer cases and we have to say with sadness that almost every case we cannot identify early results in death. As a computer engineer who wants to be useful to humanity in an issue where early diagnosis is so important, we have completed this project which will help us to think about the ones who have fallen on us at this stage. The most important factor in stepping into this project was to contribute to the early detection process, to make this process faster and to touch people's lives.

The core component of this project was the Convolutional Neural Network. We benefit from Theano and Keras libraries when building Neural Networks. We created six different models and measured the performance of these models and used the model of Convolutional Neural Network which gives the best result. We will discuss the details of these models in the following sections. We have developed an Android-based application so that the results can be better understood and the user (doctor) can use the CNN more easily.

As a result, we have established a decision-support mechanism for doctors in the early diagnosis of this disease. With this application, doctors will be able to fulfill the health of more patients with a tool that will accelerate the process of diagnosing.

TABLE OF CONTENTS

ABSTRACT ii
TABLE OF CONTENTS iii
LIST OF FIGURES iv
LIST OF TABLES iv
1. INTRODUCTION 5
1.1. Problem Description and Motivation 5
1.2. Aims of the Project 5
1.2.1. Main Aims 5
2. DEFINITION OF THE PROJECT 7
2.1. Scope of the Project 7
2.2. Success Factors and Benefits 7
2.3. Professional Considerations 8
2.3.1. Methodological Considerations / Engineering Standards 8
2.4. Literature Survey 10
3. SYSTEM DESIGN AND SOFTWARE ARCHITECTURE 11
3.1. Project Requirements 11
3.1.1. Functional Requirements 11
3.1.2. Non-functional Requirements 11
3.2. System Design 12
3.2.1. UML Use case Diagram for the main use cases 12
3.2.2. User Interface 12
3.3. Test Plan 13
3.4. Software Architecture 14
3.4.1. Data Flow 14
3.4.2. Control Flow 15
4. TECHNICAL APPROACH AND IMPLEMENTATION DETAILS 16
5. SOFTWARE TESTING 20
6. CONCLUSION AND FUTURE WORK 23
REFERENCES 24

LIST OF FIGURES

Figure 3.1. Main use case diagrams 13
Figure 3.2. Android application screenshots 13
Figure 3.3. Software Data Flow 15
Figure 3.4. Software Control Flow 16
Figure 4.2. A Convolutional Neural Network example 18
Figure 5.1. The graph shows the number of layers for each model 21
Figure 5.2. The graph shows the total training duration for each model 21
Figure 5.3. The graph shows the accuracy rates for each model 22

LIST OF TABLES

Table 3.1. Test case 1 .... **Error! Bookmark not defined.**
Table 3.1. Test case 2 .... **Error! Bookmark not defined.**
Table 3.1. Test case 3 .... **Error! Bookmark not defined.**
Table 4.1. Biological NN versus Artificial NN .... **Error! Bookmark not defined.**
Table 4.2. The 23 top-level categories for the Dermnet taxonomy **Error! Bookmark not defined.**
Table 6.1. Table indicates the existing pros and cons of the methods have been used .... **Error! Bookmark not defined.**

1. INTRODUCTION

1.1 Problem Description and Motivation

Today, one of the most important problems of the healthcare industry is that medical doctors (especially the general practitioners) are expected to diagnose on the diseases that they have little experience on. This causes many diseases to be misdiagnosed which inherently leads to malpractices. In addition, preliminary misdiagnosis also causes time and money loss. With this project, we aim to transfer the knowledge and experiences of doctors who are experts in the field to the doctors who are inexperienced through our application which has learned the decisions made by the experts and to support them in decisions made by inexperienced doctors. In this graduation project, specific focus will be on dermatologic diseases.

On the other hand, the fact that the patients who have a dermatologic disease and consult directly to the hospital in the first stage leads to excessive intensity in our hospitals which run under limited resources. This situation significantly reduces the quality of health care provided. In our project, the patients are firstly examined by general practitioners and the doctor who uses our application will be directed by the doctor to the related departments of the hospital according to the preliminary diagnosis. We will focus on solving such problems by using the application.

1.2 Aims of the Project

1.2.1. Main Aims

✓ Increase the reliability of the diagnosis. At present, 66% accuracy rate among dermatologists is increased to 72% accuracy rate with the neural network. The recognition to be created has the advantage that it is compared with past deliveries, not only with an expert, and more accurate recognition is made. In the first phase, our aim is to reach this 66% accuracy. On later phases, our new aim will be 85% accuracy.

✓ Creating a Diagnostic Repository: Today, the number of photos in a total of 4,000 [1] in 2 labelled disease classes will be tried to be raised to 10,000 labelled photos per year. To ensure this, every new diagnosis made is planned to be added to the existing repository to make a diagnostic contribution in the future.

- ✓ Reducing diagnostic cost: At present, the average money spent on diagnosis and treatment of dermatological is 926 TRY [2]. We aim to prevent unnecessary money spending by avoiding the misdiagnosis resulting in erroneous treatments.
- ✓ Evaluation from current diagnostic perspective: Be able to respond quickly (1 sec approx.) and effectively (72%) through easy learning of the mutations during the illnesses.
- ✓ Increase the speed of diagnosis: Utilizing visual data to ensure quick results from existing diagnostic procedures. To ensure that the doctor dedicates extra time to the examination by the help we provide, by reducing diagnosis time. (6 mins approx.)

1.2.2. Indirect Aims

- ✓ Reducing density of hospitals: Remove the necessity for patients to go to large hospitals for diagnosis and make them to go to alternative institutions where they can be diagnosed.
- ✓ Determining approximate risk of diagnosis: To determine the risk of the practitioner's reputation to support doctors in their decisions.
- ✓ Exchanging experience between doctors: Using the diagnoses of experienced doctors to help other doctors to make decisions.
- ✓ Determining the parameters used in disease detection: The surface area, colour, surface characteristics (rough or not)
- ✓ Determine the importance coefficient of these parameters.
- ✓ Increase patient / doctor rate: Due to the rapidity of the preliminary diagnosis, the doctor can spend more time with the patient
- ✓ Informing the patient about the risks
- ✓ Informing doctors about possible treatment methods

2. DEFINITION OF THE PROJECT

2.1 Scope of the Project

Our project is aimed at creating a preliminary diagnosis of dermatological diseases in the human body by using image processing methods of photographs taken with the help of phone camera. To achieve this goal, we are going to use Convolution Neural Networks, then alter weights in convolution layers in the direction of our purpose, and finally we will try to reach 85% of the highest accuracy level currently reached. In case of success, we would like to obtain a higher level of accuracy than current accuracy. We will design a user interface so that the result of the calculations that will be made by the network we create, will be clear to the doctor. We foresee our project phases as follows:

1. Obtaining the data
2. Implementation and training of Neural Network
3. Making Convolutional Neural Network structure suitable (adjusting weights)
4. Creating an Android application for easier understanding of results

2.2 Success Factors and Benefits

2.2.1. Measurability / Measuring Success

Such learning algorithms are not 100% accurate. This is because the number of input and output related parameters is too large, and the importance coefficients of these parameters cannot be determined precisely. Established learning systems will try to estimate these coefficients and we expect the accuracy of the algorithm to increase day by day as it increases the likelihood of correct results in the parameter coefficients as the number of inputs increases.

In the case of photographs, it is proven that the trained systems (72%) are superior to the decisions made by the dermatologists (66%) [1]. However, we would like to draw attention to the fact that these systems cannot be a decision maker in their own right. We aim to capture at least as accurate as these systems in the project. If the decisions made by the systems on the cases are at the level of 50% accuracy, they should be completely ignored and acted according to the doctor's knowledge or direction.

2.2.2. Benefits / Implications

In the present system, if a patient is injured in their skin, they have to make an appointment from one of the central hospitals and be examined there. If we present it to the system, it will be at all doctors who have access to the diagnostic or preliminary capacity of central hospitals. This will create a faster diagnosis / treatment opportunity for the patient.

Moreover, thanks to the data collected in the system repository, new diseases or different symptoms of known diseases will always be up to date. This will prevent from possible misdiagnosis or treatment and prevent from disease and waste of material and spiritual resources.

Finally, we aim to facilitate decision-making by informing doctors who are not experienced about the symptom, about similar symptoms and results.

2.3 **Professional Considerations**

2.3.1. Methodological Considerations / Engineering Standards

Since the libraries of Convolution Neural Network (CNN) algorithms are now available for the Python language, we will use python in our project. We are considering using one of the most common Convolution Neural Network libraries which are TensorFlow, Theano or Keras. We will create test cases for these libraries and decide what will be the best library for us based on the results of these test cases. After this step we will adjust on our neural network and adapt our target. We will make the neural network to work with the android application which is to be carried out in the next phases of the project.

We will go through a plan using UML diagrams to make our algorithms more effective. We will use the Agile Development Methods to stay within the plan. Since we have a software team of three people in the project phases, we are planning to follow up each other with Trello Application to make project follow-up easier. We also plan to work through a private repository on GitHub to prevent possible data loss in the development phase.

2.3.2. Societal / Ethical Considerations

Economical: Once the patient is diagnosed, the treatment process begins. This process is costly and can lead to negative results in the long term as well as misdiagnosis. With this project we aim to reduce incorrect preliminary diagnosis as much as possible at the initial phase between the patient and the doctor. In this way, we plan to prevent the wrong treatment costs and getting the wrong drugs.

Ethical: For the algorithm to work, patients need to share the image of the wound and its surrounding area with the application. In addition, the photographs that used to train the algorithm will be taken from the patients by giving information about the situation and requesting permission. Doctors who will use the application will be assured that the photographs to be used in preliminary diagnosis are taken in the application and these photographs should not be shared at anytime and anywhere.

Sustainability: In order to continuously increase the accuracy of the algorithm that used in the application, it is necessary to continuously grow the input dataset. In this context, the new input sets are taught at intervals and the final state of the algorithm is sent as an update to all user doctors.

2.3.3. Legal Considerations

Doctors need to accept contracts and rules before using practice. The contract contains several items such as the use of photographs of the illness, social media sharing, accepting the fact that the recognition is not certain, and the application used cannot be reproduced and sold to other persons etc.

2.4 Literature Survey

Along with the increase in the processing power of the machines and the data sets available, companies have begun to work on this problem in recent years. However, since it has an aspect of business dimension, it is not available for us to reach the contents. Some of the work done in the same field with us is as follows:

1. **Codella:** They are trying to diagnose Melanoma, the deadliest kind of skin cancer, by taking advantage of image processing. They make inferences using a pre-trained NN. They use the ILSVRC 2012 as the dataset. The dataset contains 900 training images and 379 test images. They keep working on ImageNet dataset which we will use later. They state that they have achieved an accuracy rate of 70.5% in their last published articles. [2]

2. **Barata:** The approach proposed by Barata et al. [3] utilizes two different methods for the detection of melanoma in dermoscopy images based on global and local features. The global method uses segmentation and wavelets, Laplacian pyramids or linear filters followed by a gradient histogram are used to extract features such as texture, shape, and colour from the entire lesion. After that, a binary classifier is trained from the data. The second method of local features uses a BOF classifier for image processing tasks (i.e. object recognition).

3. **Kawahara:** A previously trained convolution neural network was used. They have practiced on ISIC 2017 dataset. There are 2000 labelled images on 10 dermatological diseases in this dataset. It also contains 150 test images. Likewise, we are going to do as well, they carried out performance tests on AlexNet, VGG16, ResNet50, and ResNet101 frameworks. [3]

3. SYSTEM DESIGN AND SOFTWARE ARCHITECTURE

3.1. Project Requirements

3.1.1 Functional Requirements

3.1.1.1. Upload Screen: Users who are logged in must encounter this screen. On this screen, users upload the following information to the system. The system informs user about the skin disease. There is a button on this page: Take Photo Button. When the button is clicked, the camera of the phone must be opened. So, the user takes a picture of the diseased area.

3.1.1.2. Result Screen: The system must compare the captured image to the trained NN. As a result of the comparison made, top 5 skin diseases with the highest similarity and their probabilities must be reported to the user.

3.1.2. Non-functional Requirements

3.1.2.1. Usability

3.1.2.1.1. NN must be trained when the application is first installed.

3.1.3. Efficiency (Performance)

3.1.3.1. An efficient search will be performed because the database will be indexed to the user ID's

3.1.3.2. An efficient find will be performed because the Neural Network will be on local system.

3.1.4. Portability

3.1.4.1. The system will support Android platforms.

3.1.4.2. User's android systems must be more than API 26 (Android 8.0)

3.1.5. Privacy Requirements

3.1.5.1. Verification of the data will be provided by hash functions.

3.2. System Design

3.2.1. UML Use case Diagram for the main use cases

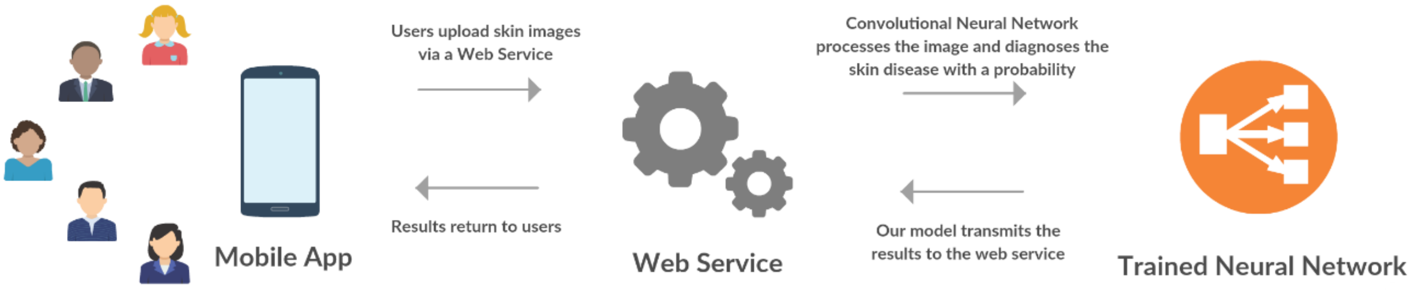

Figure 3.1. Main use case diagrams.

3.2.2. User Interface

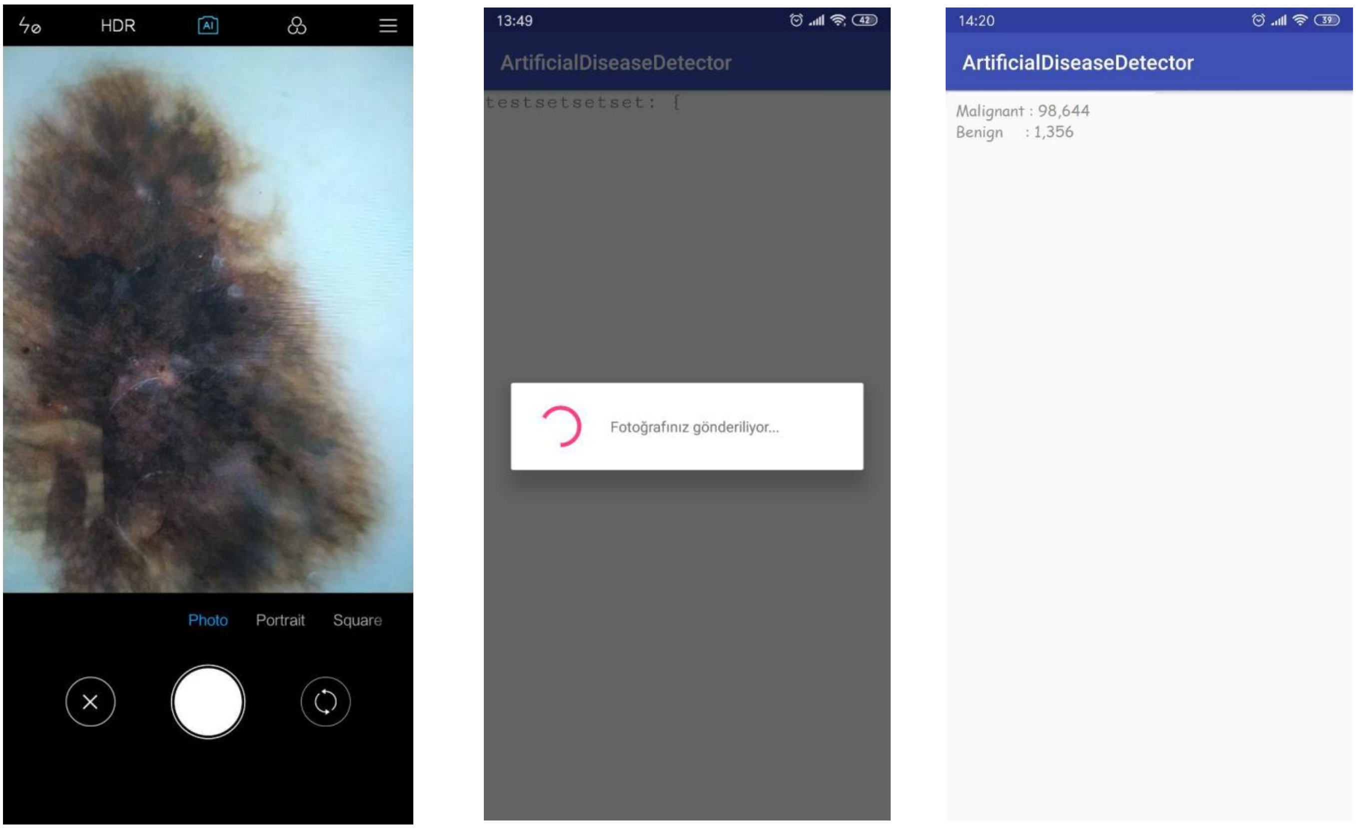

Figure 3.2. Android application screenshots.

3.3. Test Plan

This section contains a list of tests we will perform and the results we expect to achieve. The tests relate to the general use of the system.

	Test Case 1
Test Name :	Taking Photo
Procedure :	Opening the camera by pressing the take photo button and taking the photo successfully
Expected Result:	SUCCESS! Getting high quality photo.

Table 3.1. Test case 1.

	Test Case 2
Test Name :	Uploading Image
Procedure :	Upload image to the system by pressing the upload image button
Expected Result:	SUCCESS! Image uploaded successfully

Table 3.2. Test case 2.

	Test Case 3
Test Name :	Upload without Image
Procedure :	Click upload button without taking photo.
Expected Result:	ERROR!

Table 3.3. Test case 3.

3.4. Software Architecture

3.4.1. Data Flow

The core component of our system is artificial neural network. The purpose of artificial neural networks is to develop new models in order to make computers faster than human mind in a variety of ways, to develop new models and to use the processing power efficiently. At this step, we are using Convolution Neural Network. This system consists of multiple convolution layers. Each layer in the system includes an abstraction of the input data at a higher level. The output of one layer is the input of another layer associated with it. Up to 1000 convolutions with a minimum 5 can be found between layers. Inputs to the layers are calculated by the activation functions and transferred to the other layer.

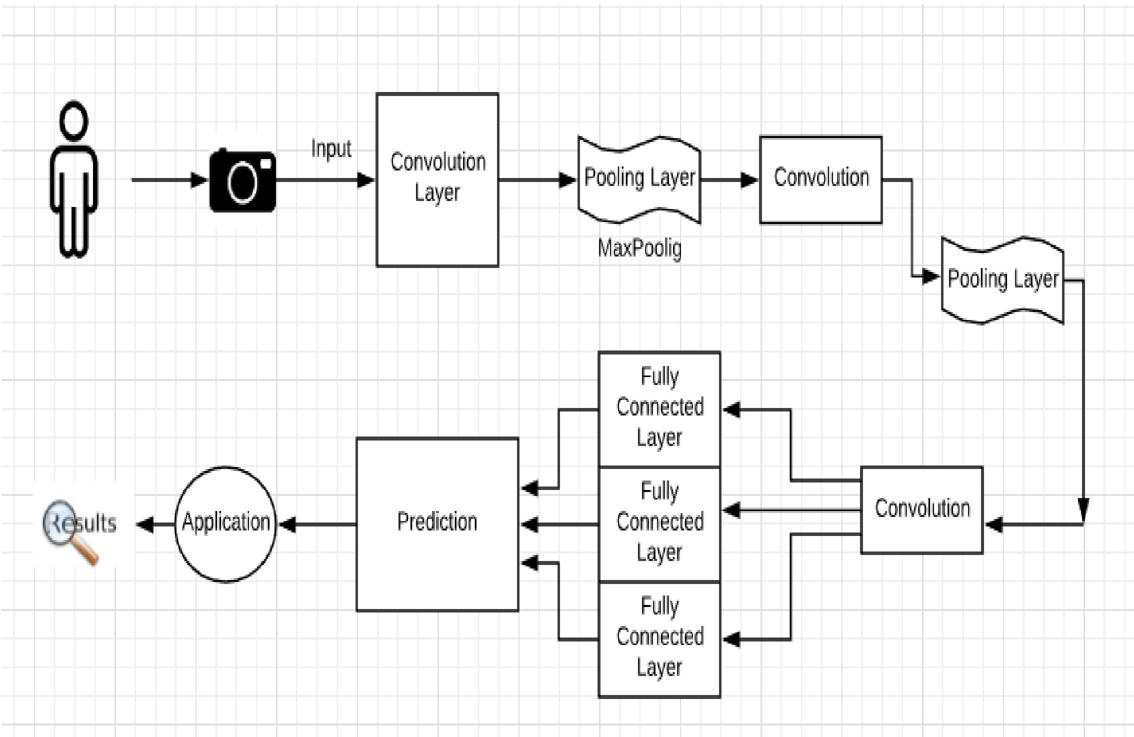

Figure 3.3. Software Data Flow.

3.4.2. Control Flow

The user will take the photo of the diseased area and upload it to the application. The photo will be input for convolutional neural network. There will be some calculations on this input image. As a result of these calculations, the disease will be estimated, and the results will be shown clearly to the user.

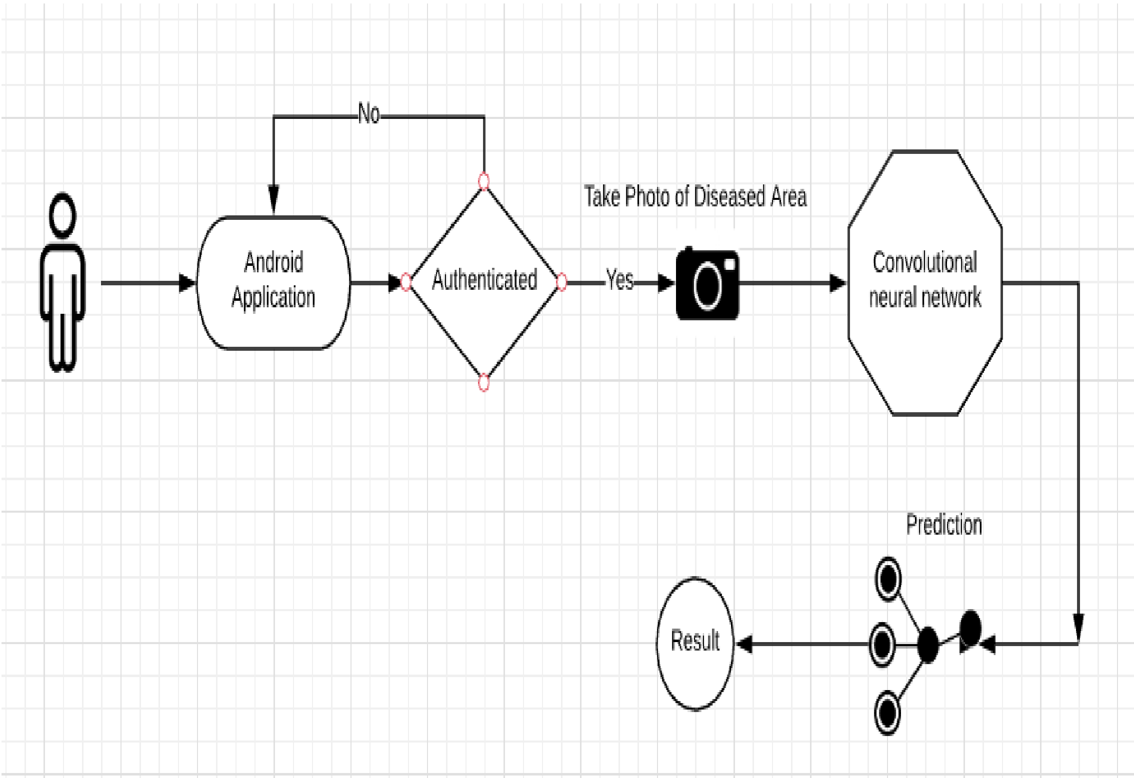

Figure 3.4. Software Control Flow.

4. TECHNICAL APPROACH AND IMPLEMENTATION DETAILS

The core component of our system is artificial neural network. The purpose of artificial neural networks is to develop new models in order to make computers faster than human mind in a variety of ways, to develop new models and to use the processing power efficiently. It is also called "Artificial Neural Network" because it is a discipline that imitates and mimics the human brain. We can visualize the structure in the following way:

In fact, artificial neural networks are systems based on biological neural networks. You can analyse comparison of this with the following table:

Biological Neural Networks	Artificial Neural Networks
Neuron	Node
Dendrite	Weights
Cell Body	Transfer and Activation Functions
Axon	Output
Synapse	Making Output to Input of Other Nodes

Table 4.1. Biological NN versus Artificial NN.

Artificial neural networks are a very appropriate system for learning algorithms used in Deep Learning.

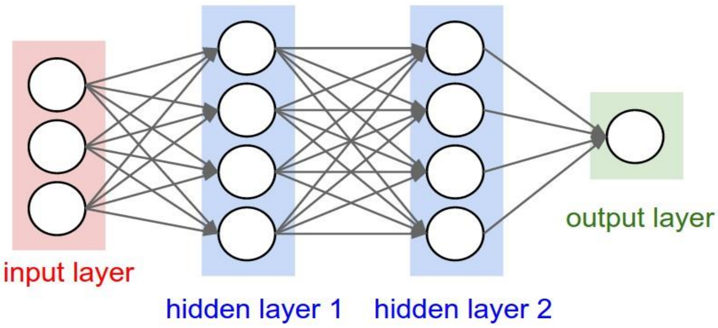

Figure 4.1. An Artificial Neural Network example [4].

One of the most useful and common methods of deep learning to process images is the Convolution Neural Network. This system consists of multiple convolution layers. Each layer in the system includes an abstraction of the input data at a higher level. The output of one layer is the input of another layer associated with it. Up to 1000 convolutions with a minimum 5 can be found between layers.

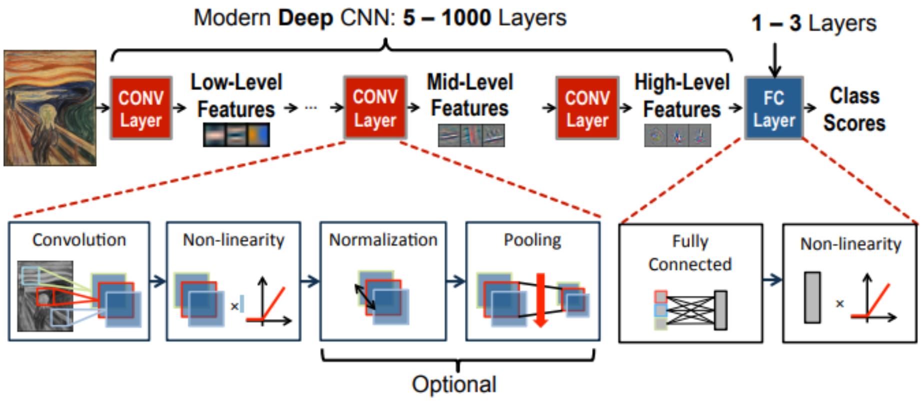

Figure 4.2. A Convolutional Neural Network example [5].

Inputs to the layers are calculated by the activation functions and transferred to the other layer. Some of the activation functions are: Linear Function, Symmetric Hard Limit, Sigmoid Function, etc.

We will use several Deep Neural Network (DNN) models in our project. We will perform a performance test among these models and work with the model which gives the optimum result. We will fine tune on a pre-trained network instead of training randomly so that we will work towards our targets. Hence, we will do Transfer Learning using a deep learning framework such as ImageNet[6] or Caffe for this. We have considered planning to use LeNet, VGG16, and GoogleNet as previously prepared models.

1. **LeNet:** This model contains 2 convolution layers and 2 fully connected layers. In the first convolutional layer, there are 6 filters and in the second one, there are 16 filters. These filters are 5x5 sizes. They use “Sigmoid Functions” as activation functions.

2. **VGG16:** There are a total of 16 layers, including 13 convolution layers and 3 fully connected layers. Each layer contains 3x3 filter images. In this model, the input layer requires 224x224 input images.

3. **GoogleNet:** This model goes deeper with 22 layers. There are parallel connections between the layers. You can apply filters in different dimensions to parallel connected layers (1x1, 3x3, 5x5, 7x7 etc.) The biggest advantage of filtering in different sizes in each layer is that input can be processed with multiple scales and provides better results. It provides 3 convolution layers and 9 inception layers between the 22 layers.

The most important challenge in this project is the variety of datasets. Today we still cannot find the desired level of labelled disease pictures to train these networks. Yet we can reach the enough number of photos to get the projects to good levels.

There are two datasets to use in our project:

1. **DERMNET:** This data set is provided by the Edinburgh Dermofit Labs database examines more than 23000 dermatological skin images in two different taxonomies available which are, bottom-level and high-level taxonomy. In bottom level, there are more than 600 different “Fine Grained Granularity” skin diseases. On the other hand, there are 23 different kinds of skin disease at the top-level.

Top-level Skin Disease Categories From Dermnet			
0. Acne and Rosacea	6. Exanthems & Drug Eruptions	12. Nail Diseases	18. Fungal Infections
1. Malignant Lesions	7. Hair Diseases	13. Contact Dermatitis	19. Urticaria
2. Atopic Dermatitis	8. STDs	14. Psoriasis & Lichen Planus	20. Vascular Tumors
3. Bullous Disease	9. Pigmentation Disorders	15. Infestations & Bites	21. Vasculitis
4. Bacterial Infections	10. Connective Tissue diseases	16. Benign Tumors	22. Viral Infections
5. Eczema	11. Melanoma, Nevi & Moles	17. Systemic Disease	

Table 4.2. The 23 top-level categories for the Dermnet taxonomy.

OLE: This dataset is provided by the New York State health department. It contains over 1300 images of skin disease. It divides skin diseases into 19 classes.

The libraries we used in our project are:

1. TensorFlow
2. Keras
3. Theano

As a result of our research, we have decided Keras library is more meritorious as comparison with other libraries. Namely:

1. Users can create their networks by following a linear sequence.
2. The functions in the library allow the user to easily create and change network layers.

The user works in an environment that gives the freedom to use additional python dependencies such as SciPy and PIL.

5. SOFTWARE TESTING

- Collecting Dataset

One of the biggest challenges in this project was the inability to find a sufficient number of photos to train our models because there are not too many institutions or people working on this issue, we have difficulty in finding data. As a result of long research, we used the data of the ISIC dataset to train our models. In this data set, there are 3500 “Benign” and “Malignant” classes that we use for training process. We have 1500 test photos which we use to test our model with this data. Our model did not encounter any of these photos during the training.

- Creating CNN models

Since we did not know how the model will perform in the project, we have designed 6 different models with different layer numbers. We measured the performance by training these models with the same data set. The layer numbers and structures of the models that we have created are listed below.

Model 1: 2 Convolution + 2 Pooling + 1 Flattening + 1 Fully Connection
Model 2: 3 Convolution + 2 Pooling + 1 Flattening + 1 Fully Connection
Model 3: 3 Convolution + 3 Pooling + 1 Flattening + 1 Fully Connection
Model 4: 4 Convolution + 4 Pooling + 1 Flattening + 1 Fully Connection
Model 5: 4 Convolution + 4 Pooling + 4 Flattening + 1 Fully Connection
Model 6: 6 Convolution + 3 pooling + 1 Flattening + 1 Fully Connection

After building the models in the above structure, we tried to see the effects of epoch parameter on the system by running with different number of epochs. In Model 1, 3 and 4 we used 10 epoch model 2 and 5 epoch model and 25 epochs in model 6.

- Evaluation of the performance of the models

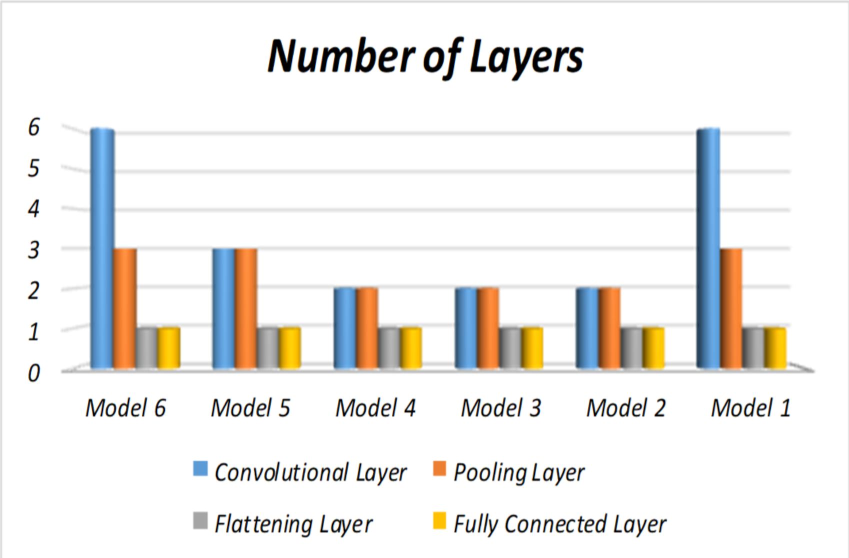

Figure 5.1. The graph shows the number of layers for each model

As shown in this graph, 6 different models were used for the most successful model selection to be integrated into our system. Some models were trained with different layer numbers.

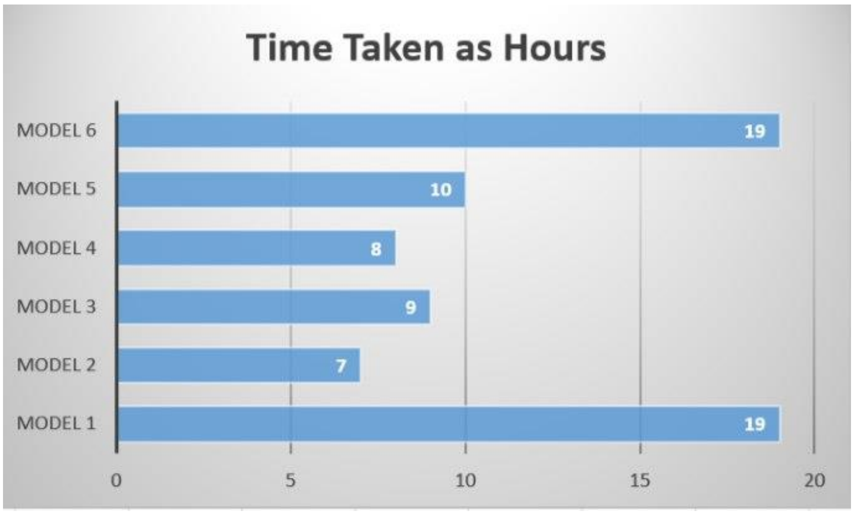

Figure 5.2. The graph shows the total training duration for each model (hours)

In the above chart, models with different layer numbers were run with the same epoch numbers. However, the epoch numbers for Model 2, 3 and 4 were different. The higher the epoch number, the more time it would take for the model to train. Another point we face is that the more trained model does not necessarily mean, the accuracy rate will increase. This leads us to the next chart.

The graph below shows the success rates achieved in 6 different models. In our second model, as shown in Figure 6.2, although the total train passing time is 7 hours, we have a very high accuracy difference between the Model 2 and the Model 5, although the duration of the Model 5 is 10 hours.

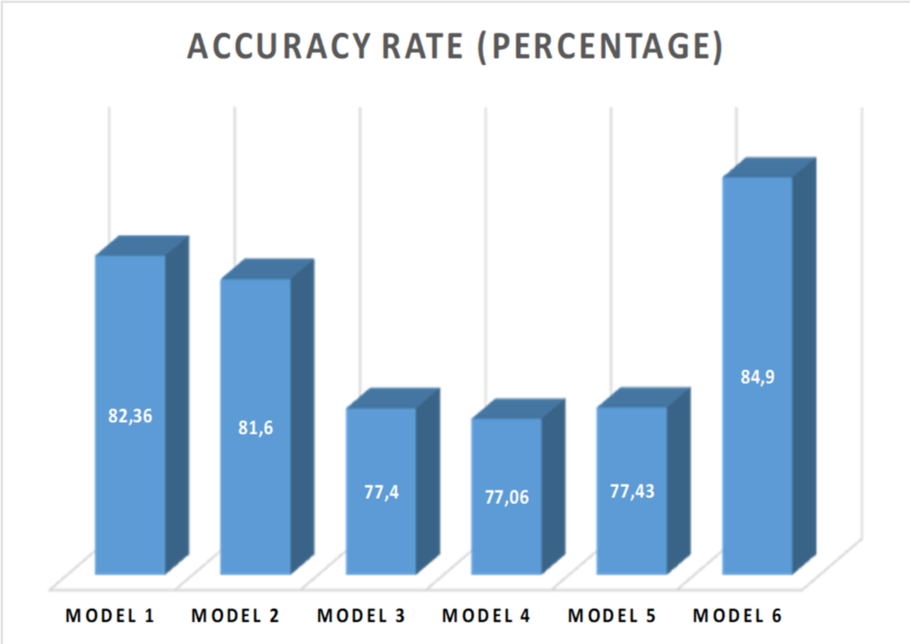

Figure 5.3. The graph shows the accuracy rates for each model

The conclusions obtained from the results of these graphs do not mean that the models which have higher number of layers will be more precise. The second point, in the same way, is to train the models over a longer period is not going to come up with a more accurate model as well. The number of layers and train time to be used for any problem vary considerably depending on the quality of the current project and the suitability of the current dataset.

- Creating Web Server

We have created a web server using Spring Boot to transfer the photos from the users to the CNN network properly. The task of this web server is to transfer the photos from the users to the local CNN server.

- Creating Android Application

We have created an Android app for users to interact with CNN server. With this application, users can see their results without having any software technical information, by taking the photo directly, sending it to the web server.

6. CONCLUSION AND FUTURE WORK

6.1 Conclusion

In a nutshell, this project consists of two parts which are the server side and the mobile application part.

The first step is getting the skin disease image from the user and uploading it to the application. Following this, via a web server, the server manages to obtain the disease image and this image is transmitted to the neural network and the model returns a prediction with a percentage of the two types of skin disease which are Benign and Malignant. After all, user(doctor) can have a general idea about the illness.

Advantages	Disadvantages
A quick response time for incoming image	Dataset availability is a must
The model can be improved continuously	CNN may give confusing results
Basic GUI design for user-end	Input image format has to be well

Table 6.1. Table indicates the existing pros and cons of the methods have been used.

6.2 Future Work

First and most importantly, the successful diagnoses rate of the CNN will be increased. Due to the dataset restriction, we were able to train the model and obtained %85 success rate. In future work, this will be improved.

Secondly, the mobile application's user interface is going to be more functional and more visual.

REFERENCES

[1] Haofu Liao, A Deep Learning Approach to Universal Skin Disease Classification, University of Rochester Department of Computer Science, USA, 2017

[2] Bilal SULA, Haydar UÇAK, Mustafa ARICA, Demographic and Clinical Characteristics of Hospitalized Patients in Dermatology Clinic, Dicle University Medical School, Dermatology Master Science, Diyarbakir, Turkey, 2015

[3] N. Codella, Q.-B. Nguyen, S. Pankanti, D. Gutman, B. Helba, A. Halpern, and J. R. Smith, "Deep learning ensembles for melanoma recognition in dermoscopy images", 2016.

[4] Stanford CS231 Course Website, Picture About Convolutional Neural Networks. http://cs231n.github.io/assets/nn1/neural_net2.jpeg

[5] Vivienne Sze, Senior Member, IEEE, Yu-Hsin Chen, Student Member, IEEE, Tien-Ju Yang, Student Member, IEEE, "Joel Emer, Efficient Processing of Deep Neural Networks: A Tutorial and Survey", 2017.

[6] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3):211–252, 2015.