**Dermatology Diagnosis System**

**Supervisor:**

**Dr. Ahmed Rafat Abas**

**Eng.Asmaa Abdalkader**

**Team members:**

Ahmed Elsayed Gomaa

Ahmed Hamdy Mohamed

Ahmed Gamal Ibrahim

Ibrahim Elsayed Mohamed

**Zagazig University, Faculty of Computers & Informatics, Computer Science Dept.**

**DECLARATION**

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Bachelor of Science in *(Dermatology Diagnosis System)*is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

# ACKNOWLEDGMENT

I would like to express my deep gratitude and appreciation to my supervisor,

**Dr Ahmed Rafat Abas**, who has been extremely supportive and quite encouraging throughout the different phases of this project. His guidance and support have always been a motivation for us to work hard. Also, special thanks to **Eng.Asmaa Abdelkader**, for her excellent support and contribution in the project.

# 

# ABSTRACT

Skin diseases are common problem among different countries and ages. Besides their painful effects they are spreading very fast to cover a large body area. The diagnosis of the skin diseases requires a high level of expertise and they are subjective to the dermatologist. Therefore, a computer-aided skin diseases diagnosis system is proposed in this paper to provide both objective and reliable solution to this problem. The system uses a deep learning model to classify the infected images. Classification performances of different architectures of the Convolutional Neural Networks are compared using the DermNet skin image database. These architectures are the AlexNet, the VGG-16, and the VGG-19. The AlexNet architecture produces the best results and is used in the proposed diagnosis system. The proposed system produces 95.7% accuracy in identifying the skin Disease from an input image for the infected area.

CONTENTS

Chapter 1 | Introduction…………………………………………………………………………....6

1.1 | Idea…………………………………………………………………………………….…7

1.2 | Overview………………………………………………………………………………….8

1.3 | Technologies Use and Methodology…………………………………………………….9

1.4 | DataSet………………………………………………………………………………….10

Chapter 2 | Literature Review…………………………………………………………………….11

2.1 | Overview ………………………………………………………………………………...12

2.2 | Related Work ………………………………………………………………………….....13

2.3 | DataSet……………………………………………………………………………………15

2.4 | Preprocessing And Image Enhancement…………………………………………………16

2.4.1 | Data Balanced ………………………………………………………………………….17

2.4.2 | Resizing Images .……………………………………………………………………….18

2.4.3 | Data Normalization ………………………………………………………………….…19

2.4.4 | Using Data Augmentation ….…………………………………………………….….....20

2.5 | Feature Extraction…………………………………………………………………….…..21

2.6 | Classification …...…………………………………………………………………….….22

Chapter 3 | Methodology And Design ..…………………………………………………………..23

3.1 | Overview ………………………………………………………………………………..24

3.2 | DataSet ……………………………………………………………………………….....25

3.3 | Learning Model ………………………………………………………………………....26

3.3.1 | Convolutional Neural Network (CNN) ……………………………………………….27

3.3.2 | Stride ………………………………………………………………………………….31

3.3.3 | Padding ……………………………………………………………………………….32

3.3.4 | Non Linearity (ReLU) ………………………………………………………………..33

3.3.5 | Pooling Layer …………………………………………………………………………34

3.3.6 | Fully Connected Layer ………………………………………………………………..35

3.4 | Architecture Of Alex net ………………………………………………………………..36

3.5 | VGG 19 ………………………………………………………………………………….38

3.6 | VGG 16 ………………………………………………………………………………….39

Chapter 4 | Planning……………………………………………………………………………....40

4.1 | Planning Phase …………………………………………………………………………..41

4.2 | SDL Four Fundamental Phases ………………………………………………………….42

Chapter 5 | System Analysis ………………………………………………………………………45

5.1 | Analysis Phase …………………………………………………………………………...46

5.2 | Interview …….……………………………………………………………………………47

Chapter 6 | System Design ………………………………………………………………………...48

6.1 | Use Cases ………………………………………………………………………………...49

4.2 | Context Diagram ….……………………………………………………………………...50

6.3 | Data Flow Diagram ………………………………………………………………………51

Chapter 7 | Implementation And Testing ………………………………………………………….53

7.1 | Integrated System Results ………………………………………………………………..54

7.2 | System Design ……………………………………………………………………………55

7.3 | Screen Shout ……………………………………………………………………………...56

7.3.1 | Front End ……………………………………………………………………………….56

7.3.1 | Back End ……………………………………………………………………………….60

Chapter 8 | Discussion And Conclusion ……………………………………………………………61

8.1 | Discussion Of Results …………………………………………………………………….62

8.2 | Conclusion …………………………………………………………………………….…..63

8.3 | Feature Work ……………………………………………………………………………...64|

Reference….. ……………………………………………………………………………………….65

## Chapter 1 INTRODUCTION

* 1. **IDEA**

In the beginning while presenting the idea to Dr. Ahmed Raft. It is a system for the detection of dermatological images. We were asked to go to a specialist in the field and the question was whether it was possible to identify dermatological images. In fact, we visited the Faculty of Medicine at Zagazig University. We arranged a meeting with the head of the dermatology department, Dr. Manal .and she agreed, with the idea and answered yes. It is possible to identify the dermatological diseases through the pictures and asked one of the specialized doctors in this field Dr. Nehal to follow up with us and ask her for help. After that we go to Dr. Ahmed Raft and agreed to the idea and from this perspective we started this project

* 1. **Overview**

The skin is the largest organ in the human body. It covers all the body and it is exposed to various environmental factors. Skin diseases become wide spread in the recent years all over the world. Many of these diseases are very dangerous such as skin cancer there are about 75% of deaths associated with skin cancer (Jerant et al., 2000).

* 1. **Technologies Use and Methodology**

Nowadays, technologies have changed our daily life in all aspects and the medical field is not an exception. Several medical systems have been developed to help both patients and doctors in different ways, starting from registration process ending with the use of technologies for diagnosing diseases (Kleinman, 1978; Lee, 2017; Foster, 1976).

This project proposes a diagnosis system that can identify skin diseases using a deep learning algorithm that is the convolutional neural networks (CNNs) from an input image. The DermNet skin image data set (DermNet, 2019) is used to train the CNN to be able to identify the 23 categories of skin diseases (Cavalcanti et al., 2010; Shoieb et al., 2016; Liao, 2016; Lopez, 2017).

Our proposed model is providing accuracy up to 95.7% using Alexnet architecture. This project aims to develop the skin diseases diagnosis system with a web site, the system is built on a Deep learning model to classify the infected images using Bag of Features Model with CNN classifier and develop a web site interface application to capture the images from users and the system will determine the type of the disease.

Although that there are some problems occurred through the development of the system starting with the data collected that are not balanced that obstacle the classification process and result nearly to the largest categories of Images, and overfitting which lead to bad test accuracy on the other hand lead to high train accuracy.

* 1. **Dataset**

The DermNet data set is one of the largest photo dermatology sources that is available publicly. It has more than 23,000 skin disease images on a wide variety of skin conditions. DermNet organizes the skin diseases biologically in a two-level taxonomy. The bottom-level contains more than 600 skin diseases in a fine-grained granularity. The top-level contains 23 skin disease classes. Each class contains a collection of the bottom-level skin diseases.



**Figure - 1**

**Chapter** 2

**LITERATURE REVIEW**

**Literature Review**:

This chapter presents some related theories, also an overview of previous skin diseases diagnosis systems and the technologies implemented in these systems are discussed.

**2.1 Overview**

Skin diseases, in the last few decades, become very common all over the world, for example number of skin cancer infections has been doubled in the past 15 years , with this extensive spreading and the severity of large number of the skin diseases arise the need for radical solutions for these diseases, the traditional diagnosis of skin diseases requires high level of expertise in the domain and a large amount of knowledge to differentiate between the large number and similar diseases, also it depends on the visual aspects of the physicians, and some of them can’t be distinguished just with the human eye and requires further tools or tests, so the traditional skin diseases diagnosis is subjective and may be unreliable, then more proper solutions are needed. This motivates researchers to propose computer aided systems for diagnosing Skin Diseases, especially with the increasing role of the deep learning and data science implementations in the medical field, different machine learning and data science techniques were implemented in those systems because of their ability to recognize patterns in objective manner.

This chapter introduce you with some of the techniques that could be implemented to develop Skin Diseases Diagnosis System based on deep learning approach, in addition provide and discuss the previous attempts to develop Skin Diseases Diagnosis Systems using deep learning or any similar techniques.

**2.2 Related work**

An algorithm is proposed to automatically distinguish celiac disease from the video capsule image using the 10-fold with an accuracy of 86.47% and LOOCV technique with an accuracy of 85.91% (Koh, 2019). Another algorithm is proposed to identify lung cancer using tree-based classifiers like Random Forest and XGBoost with Accuracy is 84% (Bhatia et al., 2019). A comparative study on diagnosing the diabetic disease using different machine learning algorithms such as decision tree, logistic regression, K-nearest neighbors, naïve Bayes, and support Vector Machine (SVM) finds out that the logistic regression gives the most accurate results that is 77% (Choudhury and Deepak, 2019). A comparative study on diagnosing the Parkinson’s disease using Decision Tree, Naive Bayes, Neural Network, Random Forests, and SVM finds out that the Random Forests has the highest diagnostic accuracy that is 99.49% (Mostafa, 2019). In diagnosing Liver cancer, the SVM with new 2-level genetic optimizer and feature selection produces classification accuracy of 88.49% (Książek, 2019). Deep convolutional neural networks (DCNNs) is used on chest radiographs for detecting tuberculosis (TB) with Accuracy of 99% (Lakhani and Baskaran, 2017). The SVM is used to identify heart disease with accuracy 94.60% (Fatima and Maruf, 2017). The naive Bayes algorithm is used to identify diabetes disease with Accuracy 95% (Fatima and Maruf, 2017). The FT Tree Algorithm is used to identify liver disease with Accuracy 97.10% (Fatima and Maruf, 2017). The RS theory is used to identify dengue disease with Accuracy 100% (Fatima and Maruf, 2017). The feed forward neural network is used to identify hepatitis disease with Accuracy 98% (Fatima and Maruf, 2017).

Several computer aided systems are developed to diagnose many types of skin diseases. A classification algorithm for identifying melanoma, which is a type of skin cancer, is developed using DermNet Data set with accuracy 91% (Mukherjee et al., 2019). This algorithm uses particle swarm optimization (PSO) and simulated annealing (SA). A CNN that is trained with DermNet Data set is used to recognize skin diseases that are produced by diabetes with accuracy of 70% and after using data augmentation the accuracy becomes 91% (Gupta, 2018). The CNN that is trained with DermNet Data set is used to identify melanoma, which is a type of skin cancer, with accuracy of 90.5% (Emre, 2008). The SVM, K-Nearest Neighbor (K-NN) and CNN that are trained with DermNet Data set are used to identify melanoma with accuracy of 94; 51% (Κατρίνη, and Chrysanthi, 2017). It is noticed that all these algorithms use training data that is not balanced, i.e., it is composed of different categories that are of different sizes. This causes bias in the learning process and results in low classification accuracy.

**2.3 Dataset**

Dermnet dataset is one of the largest photo dermatology sources that is available publicly. It has more than 23,000 skin disease images on a wide variety of skin conditions. Dermnet organizes the skin diseases biologically in a two-level taxonomy. The bottom-level contains more than 600 skin diseases in a fine-grained granularity. The top-level contains 23 skin disease classes. Each of the top-level skin disease class contains a collection of the bottom-level skin diseases, the data was divided into training dataset, validation dataset and testing dataset using the following percentages 80% training, 15% validation, 5% testing.5% validation, 5% testing.



Figure -2 Dermnet dataset

### 2.4 Preprocessing and Image Enhancement

Data is preprocessed before being fed into the convolution neural network. Keras platform has a module with image-processing tools, located at keras.preprocessing.image.

This module contains the class ImageDataGenerator, which allows Python generators to be set up. These generators turn image files on disk into batches of preprocessed tensors that are used as input to the CNN.

Figure -3

**2.4.1 Data Balanced**

Dermnet dataset is not balanced, i.e., it contains classes of different sizes. This leads to biased learning to the CNN and therefore increases the mis-classification errors. Balanced data is obtained by using data augmentation i.e., classes are made of the same size by using some of image processing operations such as rotation of different angles, zooming, shearing, height shift, width shift, and horizontal flipping. Data augmentation increases the size of the input dataset and therefore decreases the effect of overfitting while learning.

\*Image rotate (90),(180),(270)and (50)

image\_rot\_90 = image.rotate(90)

\*image flipping

image\_flip = image.transpose(Image.FLIP\_LEFT\_RIGHT)

Main Image Rotate (90) Rotate (180)

Figure -4

**2.4.2 Resizing images**

Images in the Dermnet dataset are of different sizes. Before training the CNN, these images are resized to be of fixed size that is 224x224 pixels

Figure -5 / images default size Figure -6 /fixed size 224x224 pixels

2.4.3 **Data Normalization**

Data normalization is an important step which ensures that each input parameter (pixel, in this case) has a similar data distribution. This makes convergence faster while training the network. Data normalization is done by subtracting the mean from each pixel and then dividing the result by the standard deviation. The distribution of such data would resemble a Gaussian curve centered at zero. For image inputs we need the pixel numbers to be positive, so we might choose to scale the normalized data in the range [0,1].

New Pixel= Pixel/255

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| .039 | .590 | .243 | .082 | .866 |
| .203 | 0 | .215 | .176 | .058 |
| .215 | .345 | .388 | .015 | .274 |
| .129 | .023 | .129 | .027 | .388 |
| .078 | .011 | .035 | .031 | .470 |

ImageDataGenerator(rescale=1. /255)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 10 | 15 | 62 | 21 | 221 |
| 52 | 0 | 55 | 45 | 15 |
| 55 | 88 | 99 | 4 | 70 |
| 33 | 6 | 33 | 7 | 99 |
| 20 | 3 | 9 | 8 | 120 |

Table -1 /input image Table -2 /normalization image

**2.4.4 Using data augmentation**

Overfitting is caused by having too few samples to learn from, rendering you unable to train a model that can generalize to new data. Given infinite data, your model would be exposed to every possible aspect of the data distribution at hand: you would never overfit. Data augmentation takes the approach of generating more training data from existing training samples, by *augmenting* the samples via a number of random transformations that yield believable-looking images. The goal is that at training time, your model will never see the exact same picture twice. This helps expose the model to more aspects of the data and generalize better.

In Keras, this can be done by configuring a number of random transformations to be performed on the images read by the ImageDataGenerator instance. Let’s get started with an example.

EX:

datagen = ImageDataGenerator(

rotation\_range=40,

width\_shift\_range=0.2,

height\_shift\_range=0.2,

shear\_range=0.2,

zoom\_range=0.2,

horizontal\_flip=True)

These are just a few of the options available.

Let’s quickly go over this code:

\* rotation\_range is a value in degrees (0–180), a range within which to randomly

rotate pictures.

\* width\_shift and height\_shift are ranges (as a fraction of total width or

height) within which to randomly translate pictures vertically or horizontally.

\* shear\_range is for randomly applying shearing transformations.

\* zoom\_range is for randomly zooming inside pictures.

\* horizontal\_flip is for randomly flipping half the images horizontally—relevant

when there are no assumptions of horizontal asymmetry (for example,

real-world pictures).

\* fill\_mode is the strategy used for filling in newly created pixels, which can

appear after a rotation or a width/height shift.

### 

### Feature Extraction

Feature extraction is the process of detecting a certain features of interest within an image in order to use it for further processing, generally, this is a critical step in the image processing solutions because it marks the transition between pictorial to non-pictorial data representation, and the resulting outcome of the feature extraction process can be subsequently used as an input to a further pattern recognition or classification technique in order to label, classify or recognize the content of the input image .

The process of finding the corresponding correlation between two images of the same scene acquired by the same or different sensors, at different time or from different viewpoint is called image registration, there are several steps involve in the registration process:

Feature Detection: Salient and distinctive objects in both reference and sensed images are detected.

Feature Matching: The correspondence between the features in the reference and the sensed image established.

Transform Model Estimation: The type and parameters of the so-called mapping functions, aligning the sensed image with the reference image, are estimated.

Image Resampling: The sensed image is transformed by means of the mapping functions.

### Classification

Data mining is the process of extracting patterns from data, its referred to as knowledge discovery from databases, classification techniques are widely used in data mining to classify data among various classes [3], they are considered as a type of supervised learning when both input features and the target output are represented during the learning process, this technique typically used to predict group membership for data instances.

In Skin diseases diagnosis systems, although the features that extracted from the image represent the image in a simpler way, but it will be useful to implement the classification techniques in the prediction of the specific disease (predicted output) which has specific group of features (inputs).

**Chapter 3: METHODOLOGY AND DESIGN**

* 1. **Overview**

This chapter presents each step of the design and implementation of the skin diseases diagnosis system and discuss the methods used in each step, and provide figures and tables from the implementation process for more explanation, the chapter is divided into three main parts the first one is discussing how the data has been gathered, the second one is discussing the classification model design and implementation and the third part is considering the system development and integration.

* 1. **Dataset**

Generally, collecting data that fit your application is one of the most difficult steps in developing a machine learning application, so for developing the Skin Diseases Diagnosis system based on captured images, the required data are images with a labelled classes of skin diseases, so we had two choices either to collect the images manually from hospitals, healthcare centres and individual patients or using online resources for skin diseases images database.

For the manual collection of the data, we face two major problems to apply this option, the first one is that there is lack of data collection and documentation in most of the local hospitals, neither digital data nor hard copies of the data, the second problem is that if there is a data available, it’s hardly reachable because it’s considered private for the hospitals and require permissions from the local health authorities, also the data is not sufficiently enough for the training of the model, not well prepared and require a lot of work to be ready for usage, so we choose to search for another option to collect the data.

For the online resources option, there are a lot of researches are done to obtain resources for skin diseases diagnosis images. DERMNET.COM was the dominant resource for the system data, it is one of the largest dermatology photos resource that is available publicly. Although it has more than 23,000 skin disease images on a wide variety of skin conditions, but there was not direct way to download a whole class of diseases at once, so we were forced to download each single image every time. In contrast this data couldn’t be downloaded at once, we were forced to download each image individually.

We divide our dataset to training, validation and testing

First step we increase number of images and make it balanced by using image processing ,For the training of our system, 1540 images for each classes, but they were signed with watermark in the middle of the image, For the validation of our system , 400 images for each classes, ,For the testing of our system 1150 images .

### Learning Model

Convolution Neural Networks was selected to be the model to train our Skin Diseases Diagnosis System’s data.

* + 1. **Convolutional Neural Networks (CNN):**

Is the best Algorithm for Image Classification in the Next We will discuss this Algorithm.

CNNs belong to feed forward neural networks where a signal flow through the network without forming cycles or loops.

The CNN (LeCun et al., 1989) is a category of Neural Networks that has proven very effective in areas such as image recognition (Shin et al., 2016; Tajbakhsh et al., 2016; Roth et al., 2014; Dou et al., 2016), classification (Setio et al., 2016; Anthimopoulos, 2016) and segmentation (Ronneberger et al., 2015; Chen et al., 2016).

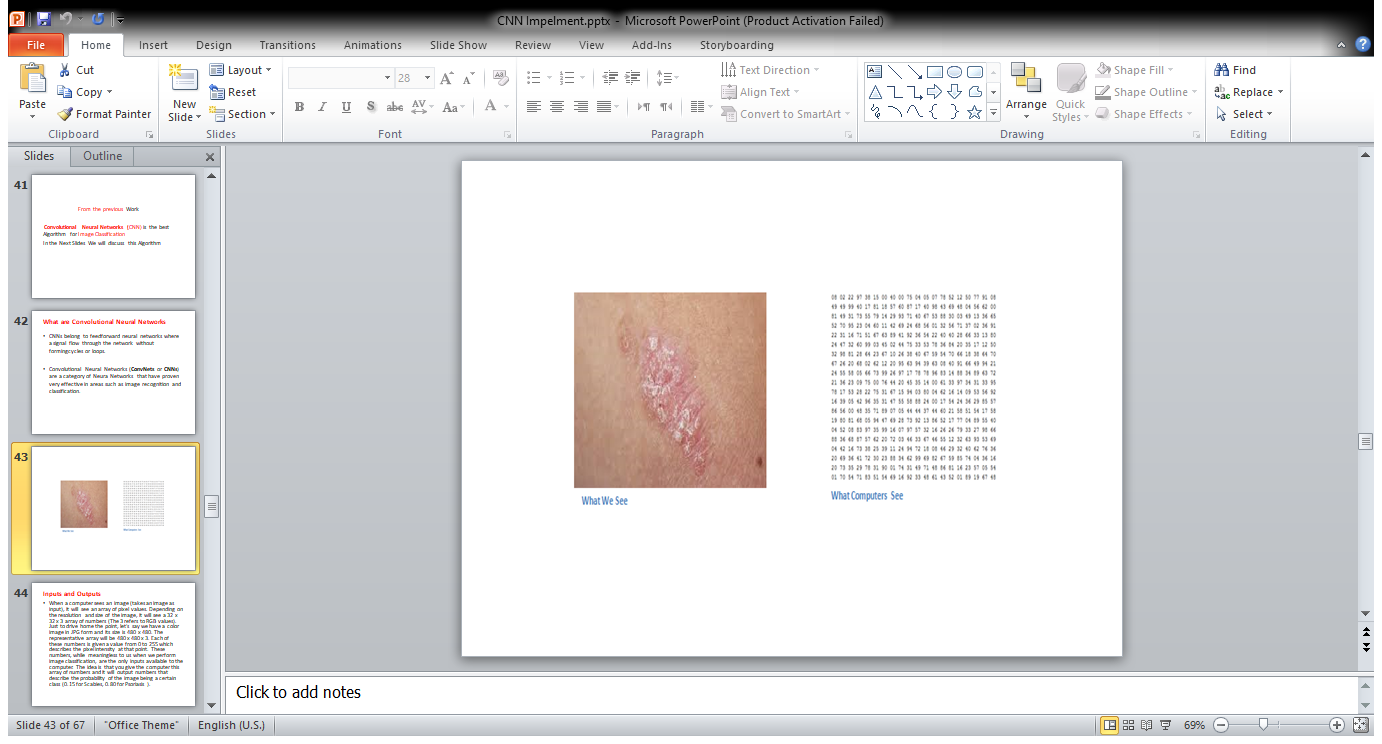


Figure -7

Each of these numbers is given a value from 0 to 255 which describe the pixel intensity at that point. These numbers, while meaningless to us when we perform image classification, are the only inputs available to the computer.  The idea is that you give the computer this array of numbers and it will output numbers that describe the probability of the image being a certain class (0.15 for Scabies, 0.80 for Psoriasis). according to figure 4.

CNNs classifiers take an input image, process it and classify it under certain categories (e.g., Scabies, Psoriasis). Computers see an input image as array of pixels. Based on the image resolution, it will see h x w x d (h = Height, w = Width, d = Dimension). For example, an image of 6 x 6 x 3 array of matrix of RGB (3 refers to RGB values) and an image of 4 x 4 x 1 array of matrix of gray scale image.

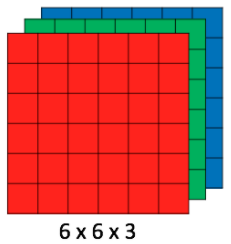


Figure -8

In deep learning CNN models each input image passes through a series of convolution layers with filters (Kernals), pooling, fully connected layers (FC) and an output layer that applies a softmax function to classify an input image with probabilistic values between 0 and 1. Figure 6 shows a complete flow of CNN to process and classify an input image

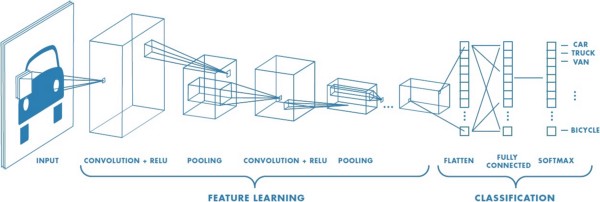


Figure -9

Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. Convolution of an image with different filters can perform operations such as edge detection, blurring, and sharpening.

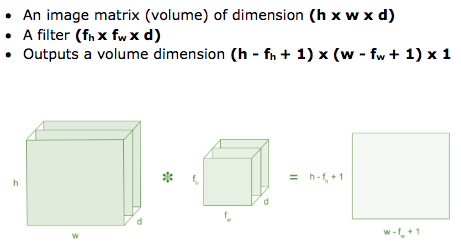


Figure -10

* Consider a 5 x 5 whose image pixel values are 0, 1 and filter matrix 3 x 3 as shown in below

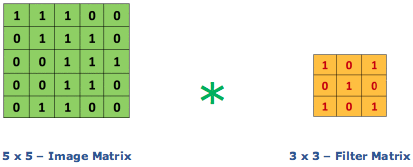


Figure -11

Then the convolution of 5 x 5 image matrix multiplies with 3 x 3 filter matrix which is called **“Feature Map”** as output shown in below

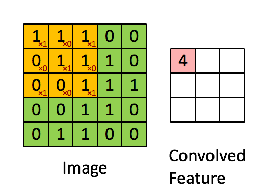


Figure -12

Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying filters. The below example shows various convolution image after applying different types of filters (Kernels)

* + 1. **STRIDE**

Stride is the number of pixels shifts over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on. The below figure shows convolution would work with a stride of 2

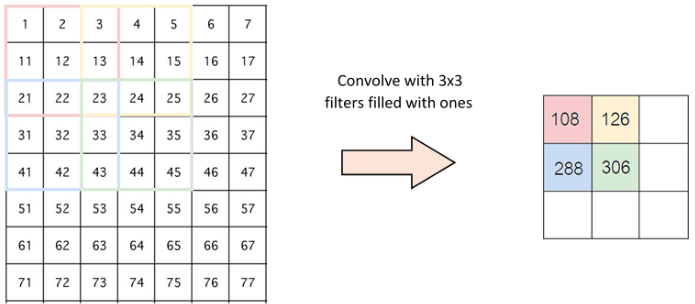


figure -13

* + 1. **PADDING**

Sometimes filter does not fit perfectly fit the input image. We have two options:

* Pad the picture with zeros (zero-padding) so that it fits
* Drop the part of the image where the filter did not fit. This is called valid padding which keeps only valid part of the image.
  + 1. **Non-Linearity (ReLU)**

ReLU stands for Rectified Linear Unit for a non-linear operation. The output is ***ƒ(x) = max (0, x).***

Why ReLU is important: ReLU’s purpose is to introduce non-linearity in our ConvNet. Since, the real world data would want our ConvNet to learn would be non-negative linear values.

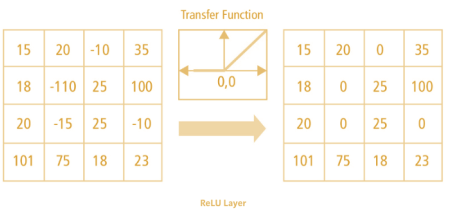


Figure -14

There are other non linear functions such as tanh or sigmoid can also be used instead of ReLU. Most of the data scientists uses ReLU since performance wise ReLU is better than other two.

* + 1. **Pooling Layer**

Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or downsampling which reduces the dimensionality of each map but retains the important information. Spatial pooling can be of different types such as:

* Max Pooling
* Average Pooling
* Sum Pooling

**Max pooling**: take the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.

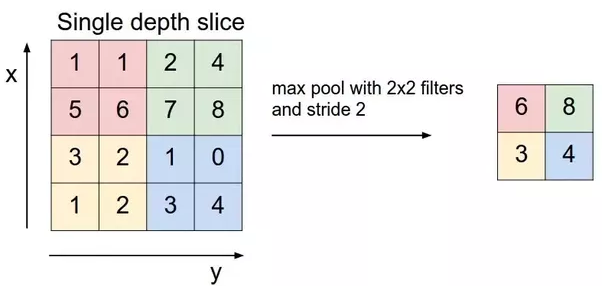


Figure -15

* + 1. **Fully Connected Layer**

The FC layer takes as input the feature map matrix that is extracted from previous layers and combines these features together to create a model. Finally, the output layer that uses a softmax activation function is used to classify the input image into one of the output classes

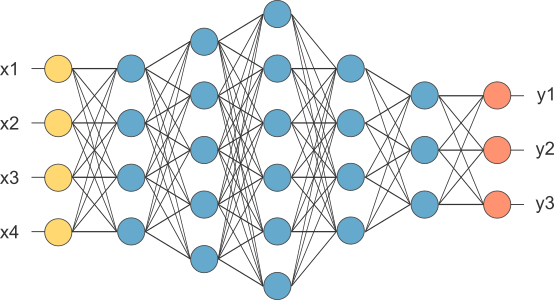


Figure -16

**This project compares three different architectures of the CNN in classification task using the dement dataset. These architectures are Alexnet, VGG 19, and VGG 16.**

* 1. **Architecture of Alexnet**

In 2012, the AlexNet architecture is proposed (Krizhevsky et al., 2012) and wins the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC). AlexNet consists of 5 convolutional layers, some of which are followed by max-pooling layers and 3 fully-connected layers with a final softmax one. The first convolutional layer filters the 224×224×3 input image with 96 kernels of size 11×11×3 with a stride of 4 pixels. The second convolutional layer takes the output of the first convolutional layer and filters it with 256 kernels of size 5×5×48. The third, fourth, and fifth convolutional layers are connected to each other without any intervening pooling or normalization layers. The third convolutional layer has 384 kernels of size 3×3×256 connected to the normalized/pooled outputs of the second convolutional layer. The fourth convolutional layer has 384 kernels of size 3×3×192, and the fifth convolutional layer has 256 kernels of size 3×3×192. The fully-connected layers have 4096 neurons.

the third fully connected layers are changed to have 23 neurons, which is the number of categories in the dermnet dataset. This architecture produces accuracy of 95.7% using 50 epochs of training.

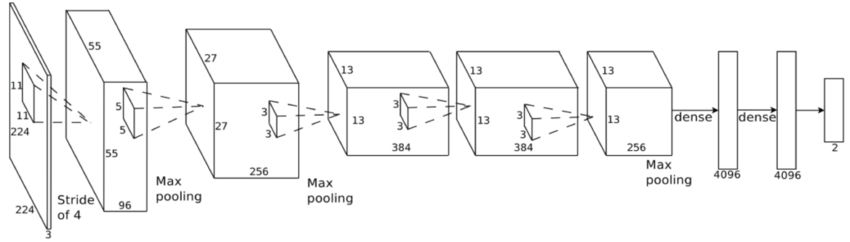


Figure -17 /Architecture of Alex net

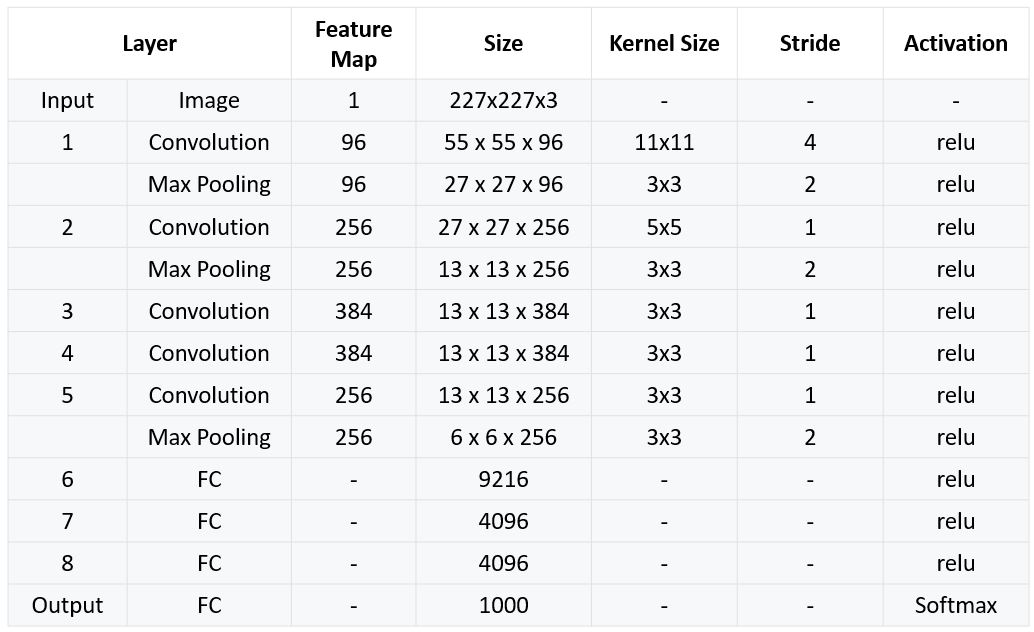


Figure -18 /layers of Alex net

* 1. **VGG 19**

Architecture of VGG19 (Simonyan and Andrew, 2014) is a CNN architecture that is used for classification. It is composed of five building blocks as shown in figure 16. The first and second building blocks are composed of two convolutional layers and one pooling layer. In the third and fourth blocks four convolutional layers and one pooling layer and four convolutional layers in the last block. The 3x3 sized filters with stride of 1 are used in all convolutional layers. In the first building block, 64 filters are oncatenated, 128 filters in the second block, 256 filters in the third block, 512 filters in the fourth and fifth blocks are used. For the pooling layers, the 2x2 sized max-pooling was done with the stride of 2. The ReLU is used asr activation function. For easier initialization and faster training, batch normalization is performed at each block. To prevent overfitting, dropout regularization is processed between fully-connected layers. In the prediction block, 4096 channels are divided into 10 classes using softmax function. This model produces classification accuracy of 94% with 50 training epochs using the dermnet dataset.

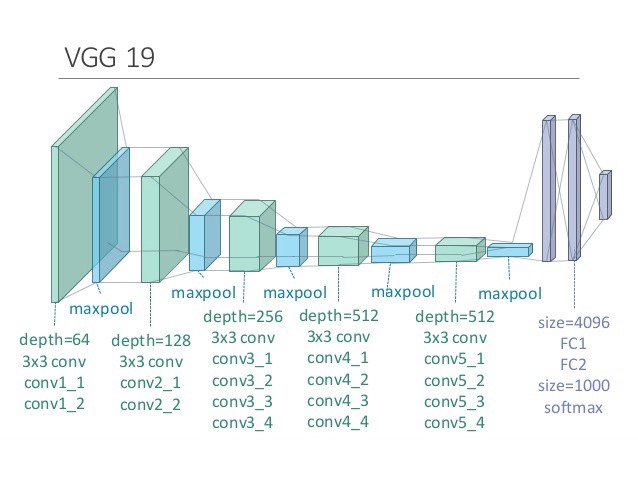


Figure -16 /Architecture of VGG 19

* 1. **VGG 16**

Architecture of VGG16 (Simonyan and Andrew, 2014) is used for classification. It is composed of five building blocks as shown in figure 17. The first and second building blocks are composed of two convolutional layers and one pooling layer. In the third and fourth blocks three convolutional layers and one pooling layer and three convolutional layers in the last block. The 3x3 sized filters with stride of one are used in all convolutional layers. In the first building block, 64 filters are concatenated, 128 filters in the second block, 256 filters in the third block, 512 filters in the fourth and fifth blocks are used. For the pooling layers, the 2x2 sized max-pooling was done with the stride of two. The ReLU is used as activation function. For easier initialization and faster training, batch normalization is performed at each block. To prevent overfitting, dropout regularization is processed between fully-connected layers. In the prediction block, 4096 channels are divided into 10 classes using softmax function. The model produces classification accuracy of 94% with 50 training epochs using the dement dataset.

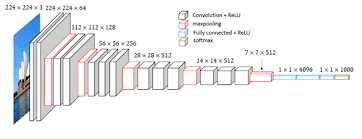
****

Figure -20 /Architecture of VGG 16

Chapter 4

Planning

**4.1 PLANING PHASE**

The systems development life cycle (SDLC):is the process of determining how an information system(is) can support business need, designing the system, building it, and delivering it to users

# 

# Figure -21

**4.2 The SDLC four fundamental Phases**

* **Planning**

In this phase we will discuss why we built this system and how will this system help all patients that have a Skin Disease

* **Analysis**

The analysis phase will explain how will use the system, what the system will do, and where and when it will be use, and also, we will explain how the data was collected.

Study and analyze the problem, causes, and effects. Then, identify and analyze the requirements that must be fulfilled by any successful solution.

* **Design**

The design phase decides how the system will operate in terms of the hardware ,software ,and network infrastructure that will be will in place the user interface ,result of diagnose of disease ,and data set that we used.

* **Implementation**

The final phase of (SDLC) is the implementation phase, during which the system.

Is actually built (or purchased, in the case of a packaged software design and design).

This is the phase that usually gets the most attention, because for most system it is the longest most expensive single part of the development process.

**PLANNING PHASE**

This phase is the fundamental process of understanding why an information system should be built, and determining how the project team will go about building it.

1. **Background project**

Our system helps patients and doctors to Diagnosis of dermatology

1. **Executive Summary**

a-by using Deep learning to train our system

b-by using python language to write our code in Deep learning and connect between design and develop by using flask python framework c-by using HTML, CSS and Java Script to make our Design

1. **Project Objectives**

Help patients diagnose diseases by uploading a picture of the system. And it gives them a diagnosis of their diseases

1. **Project Scope**

This web site allow patient to uploading a picture to system. and receive diagnosis from it

1. **Project Constraint**.

Selecting the best algorithm to provide the best accuracy of our model.

1. **Project Success**

After finishing the project, we get accuracy 95.7%.

1. **Project Benefits**

Help patients diagnose diseases.

Helping doctors in hospitals diagnose dermatology, which leads to reduce congestion in hospitals.

Help doctors in hospitals by not requiring the examination of all patients, but patients who suffer from serious diseases, which leads to the accuracy of the examination and reduce pressure on doctors

# Chapter 5

# SYSTEM ANALISIS

1. **ANALYSIS PHASE**

The analysis phase answers the questions of *who* will use the system, *what* the system will do, and *where* and *when* it will be used. And is the process of studying procedure or business in order to identify its goal and purposes and create system and procedures that will achieve them in an efficient

1. **Interview**

At the beginning of work in this project we need some information and data. By way of know we can recognize the skin diseases by using pictures or not. And collect a large data for skin diseases for train our system.

Chapter 6

System Design

* 1. **Use Case**

Use cases are a means of expressing user requirements**.**

A use case represents how a system interacts with its environment by illustrating the activities that are performed by the users and the system’s responses

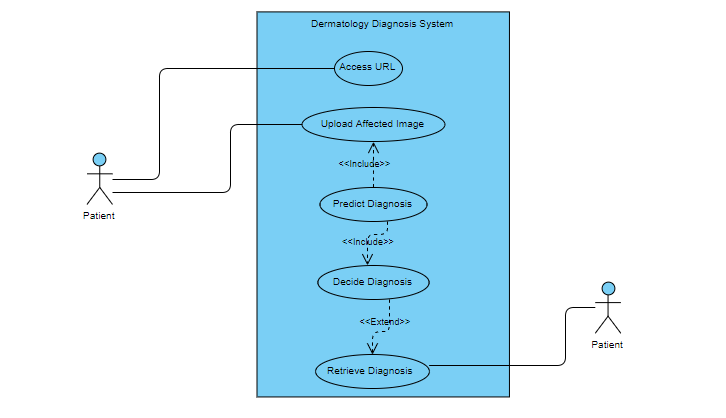


Figure -22

* 1. **Context Diagram**

Context diagram "in software engineering and system engineering" a diagram that defines the boundary between the system, or part of a system, and its environment, showing the entities that interact with it .this diagram is a high level view of a system.

Context diagram s show a system often software-based, as a whole and its inputs and outputs from/to external factor.

It representing all external entities that may interact with a system such our project diagram which pictures dermatology diagnosis system at the center, with no details of its interior structure, surrounded by all its interacting system, environments and activates.

# 

Figure -23

**6.3 Data Flow Diagram**

A data flow diagram (DFD) illustrate how data is processed by a system in terms of inputs and outputs .As its name indicates its focus is on the flow of information ,where data comes from , where it goes and how its gets stored .

**Data Flow Diagram Levels**

A context diagram is a top level (also known "level 0") data flow diagram.it only contains one process node ("process 0")that generalizes the function of the entire system in relationship to external entities.

The first level in DFD shows the main processes within the system.

Each of this process can be broken into further processes until you reach pseudo code.

# Level 0:

# 

# Figure -24

# level:1

# 

# Figure -25

# Chapter 7 Implementation And Testing

# Implementation and Testing

The learning model of the system is built based on Alexnet Architecture of CNN, The final result of the learning model of the system is successfully classify the images of the skin diseases of classes with cross validation accuracy of 95.7% .

### Integrated System Results

The skin diseases diagnosis system is successfully built with all the specified functionalities, giving the expected outcome at each step, the data successfully flow between client and server without problems.

The image is captured and is sent to the server successfully, the image information is recorded in the database, then the system script read the data immediately and load the image and perform the classification with the pre-trained model, then writing the result back to the database, then the web system read the result directly from the database and print it to the user.

The system also performs the task with a good performance, an addition its easily used requiring uncomplicated configurations to be used, so it’s a user-friendly application.

* 1. **System Design**

In this part, we discuss the overall system architecture, and explain in details the design and development process of each part of the system, and the tools and methods used to do that.

The skin diseases diagnosis system mainly consists of two parts that represents the server side and the client side each one contains a separate application and linked together over a shared network. The server side is a Flask Web Framework application that contains the main Deep learning model which implements the training and classification task. The client side is a Web application that acts as an interface to the Flask application, its main task is to receive the input from the user and pass it to the server and return the output of the server. The two sides are connected together through Flask server and Client using HTTP protocol.

* 1. **screen shout**

**7.3.1 front end:**

We use in the design HTML, CSS, JAVASCRIPT and BOOTSTRAB

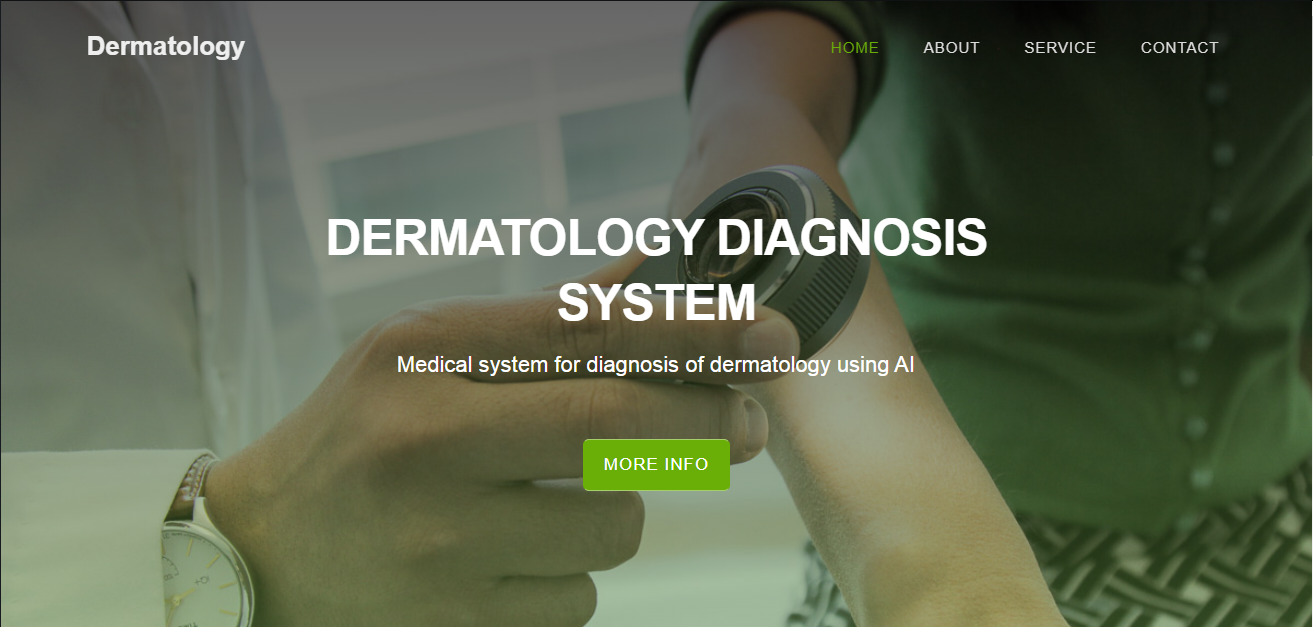


Figure -26 / screenshot1

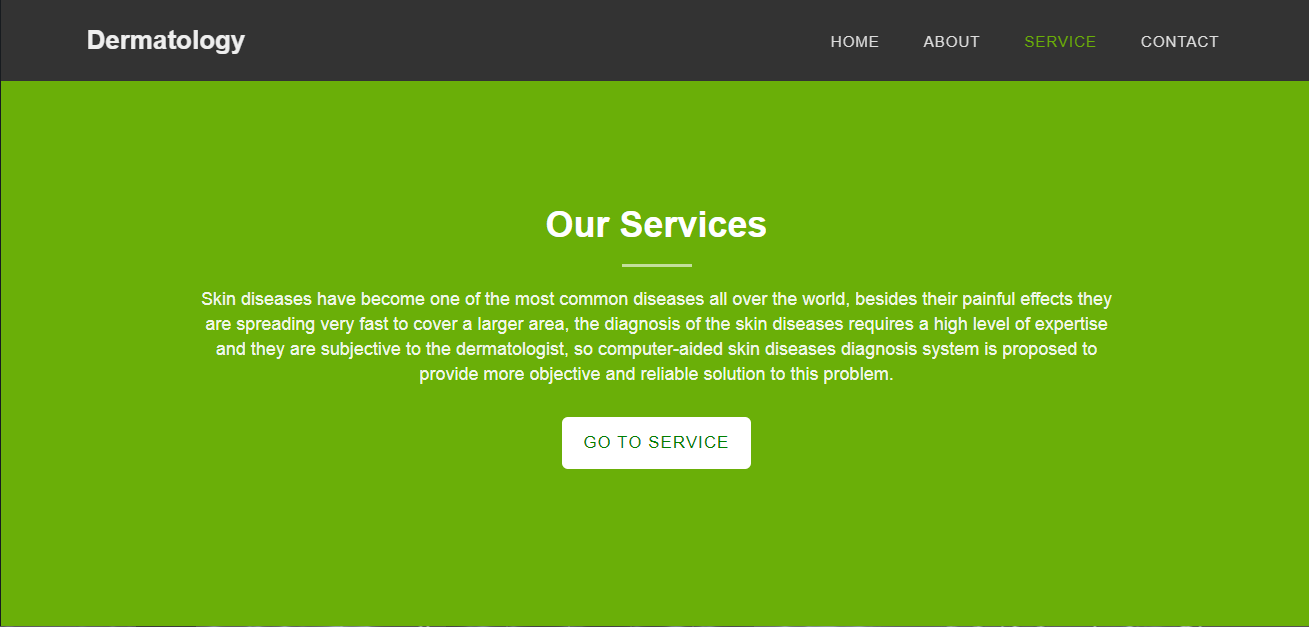


Figure -27 /screenshot2

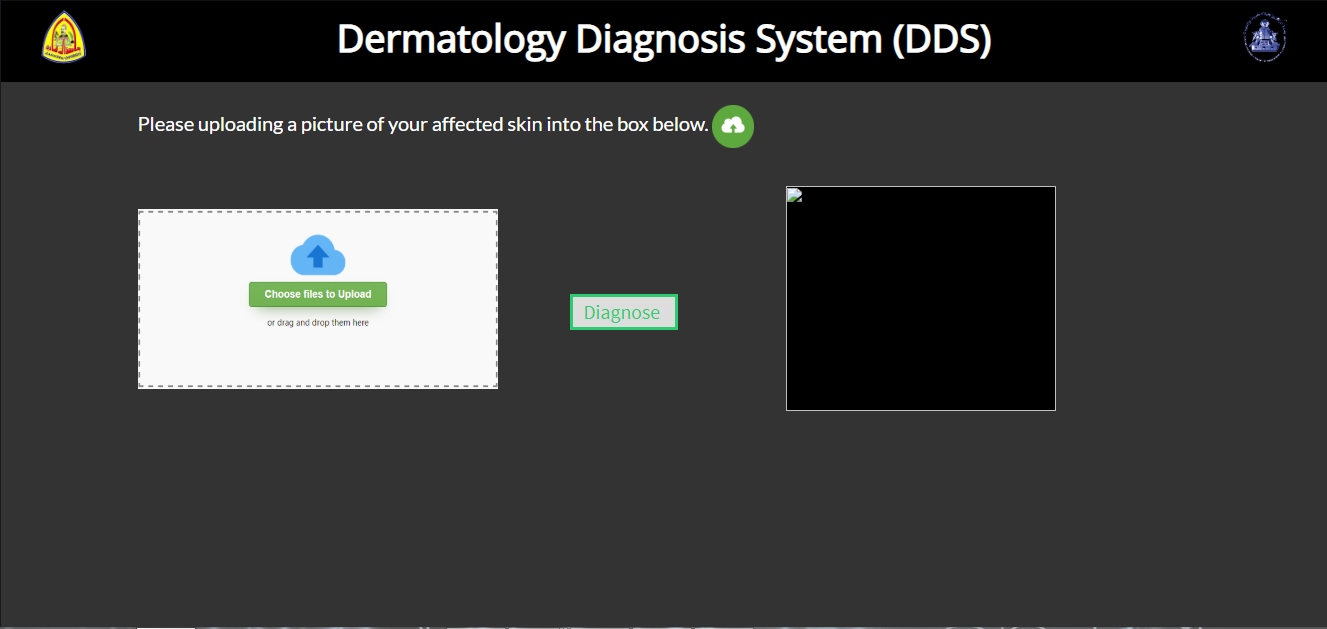


Figure -28 /screenshot3

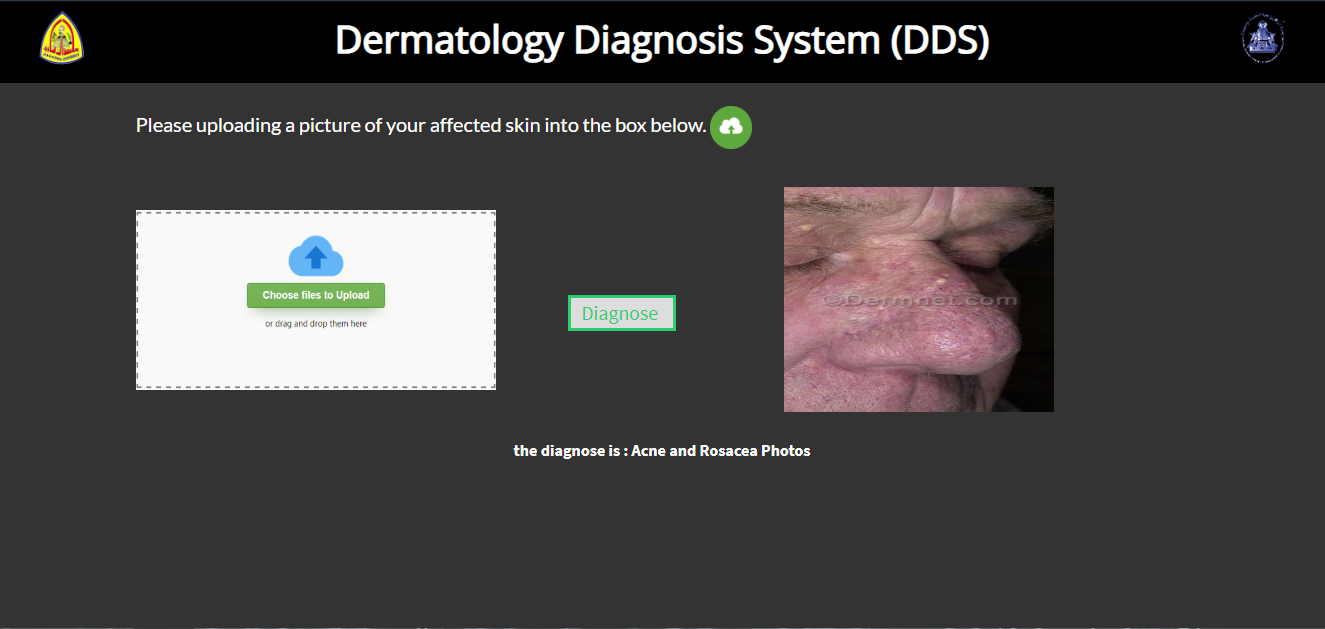


Figure -29 /screenshot4

* + 1. **Back end**

Using python framework(flask), and machine learning libraries the keras platform (Gulli and Sujit, 2017) with the tensorflow backend (Abadi, 2016) is used in the experiments. All architectures compared in this paper are trained with the DermNet data set using 50 training epochs

# 

# Chapter 8

# Discussion and CONCLUSION

* 1. **Discussion of results**

Table 3 shows that the AlexNet architecture of the CNN produces the best training and testing classification accuracy with the DermNet dataset among the different architectures compared. The Alexnet is composed of 5 convolutional layers meanwhile the VGG-16 and the VGG-19 are composed of 16 and 19 convolutional layers, respectively. This allows the AlexNet to extract less features from the images than both of the other architectures and therefore decreases the overfitting effect. This in turn increases the generalization ability of the AlexNet and in turn the classification accuracy

|  |  |  |
| --- | --- | --- |
| **CNN Architecture** | **Training accuracy** | **Testing accuracy** |
| **AlexNet** | **95.7%** | **95.7%** |
| VGG-16 | 94% | 93.6% |
| VGG-19 | 94% | 93.9% |

Table 3- Compare between architecture of CNN

* 1. **Conclusion**

Diagnosis of skin diseases requires the dermatologist to have a high level of expertise. Therefore, a computer-aided skin diseases diagnosis system that uses a deep learning model to classify the skin images is proposed. The Convolutional Neural Networks prove high efficiency as a deep learning tool in image classification. This tool is used in the proposed diagnosis system in this paper to provide fast and reliable skin disease diagnosis. After comparing three different architectures of the CNN that are AlexNet, VGG-16 and VGG-19, the AlexNet architecture produces the best classification performance with the DermNet skin disease image database to test the 23 categories of skin diseases. Therefore, the AlexNet architecture is used in the proposed diagnosis system for skin diseases.

* 1. **Future works**

1. **Create a mobile app.**
2. **increase the accuracy by getting more dataset.**
3. **Develop the architecture**
4. **Add new feature to app like treatment.**
5. **Convert the project to Arabic.**

**References**

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)* (pp. 265-283).
2. Anthimopoulos, Marios, et al. "Lung pattern classification for interstitial lung diseases using a deep convolutional neural network." *IEEE transactions on medical imaging* 35.5 (2016): 1207-1216..
3. Balch, C. M., Buzaid, A. C., Soong, S.J., Atkins, M. B., Cascinelli, N., Coit, D. G., Fleming, I. D., Gershenwald, J. E., Houghton, A., Kirkwood, J. M. (2001). “Final version of the american joint committee on cancer staging system for cutaneous melanoma”, *Journal of Clinical Oncology*, 19(16), 3635–3648.
4. Bhatia, Siddharth, Yash Sinha, and Lavika Goel. (2019). "Lung Cancer Detection: A Deep Learning Approach", *Soft Computing for Problem Solving*. Springer, Singapore, 699-705.
5. Cavalcanti, P. G., Scharcanski, J., & Lopes, C. B. (2010, November). Shading attenuation in human skin color images. In *International Symposium on Visual Computing* (pp. 190-198). Springer, Berlin, Heidelberg.
6. Chen, H., Qi, X., Yu, L., & Heng, P. A. (2016). DCAN: deep contour-aware networks for accurate gland segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 2487-2496).
7. Choudhury, Ambika, and Deepak Gupta. (2019). "A Survey on Medical Diagnosis of Diabetes Using Machine Learning Techniques", *Recent Developments in Machine Learning and Data Analytics*. Springer, Singapore, 67-78.
8. DermNet dataset. (May 2019). <http://www.dermnet.com>
9. Dou, Q., Chen, H., Yu, L., Zhao, L., Qin, J., Wang, D. & Heng, P. A., (2016). "Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks", *IEEE transactions on medical imaging*, 35(5), 1182-1195.
10. Emre Celebi, M., Hassan A. Kingravi, Hitoshi Iyatomi, Y. Alp Aslandogan, William V. Stoecker, Randy H. Moss, Joseph M. Malters et al. "Border detection in dermoscopy images using statistical region merging." *Skin Research and Technology*14, no. 3 (2008): 347-353.
11. Fatima, Meherwar, and Maruf Pasha. "Survey of machine learning algorithms for disease diagnostic." *Journal of Intelligent Learning Systems and Applications* 9, no. 01 (2017): 1.
12. Foster, George M. "Disease etiologies in non‐western medical systems." *American Anthropologist* 78.4 (1976): 773-782.
13. Freedberg, K. A., Geller, A. C., Miller, D. R., Lew, R. A. and Koh, H. K. (1999). “Screening for malignant melanoma: a cost-effectiveness analysis”, *Journal of the American Academy of Dermatology*, 41(5), 738–745.
14. Gulli, Antonio, and Sujit Pal. (2017). Deep Learning with Keras, Packt Publishing Ltd.
15. Gupta, Tanu, Satyam Saini, Aradhya Saini, Shrutika Aggarwal, and Ankush Mittal. "A Deep Learning Framework for Recognition of Various Skin Lesions due to Diabetes." In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 92-98. IEEE, 2018.
16. Jerant, A. F., Johnson, J. T., Sheridan, C., Caffrey, T. J. (2000). “Early detection and treatment of skin cancer”, *American family physician*, 62(2), 357–386.
17. Kleinman, Arthur. "Concepts and a model for the comparison of medical systems as cultural systems." *Social Science & Medicine. Part B: Medical Anthropology* 12 (1978): 85-93.
18. Koh, J. E. W., Hagiwara, Y., Oh, S. L., Tan, J. H., Ciaccio, E. J., Green, P. H., ... & Acharya, U. R. (2019). Automated diagnosis of celiac disease using DWT and nonlinear features with video capsule endoscopy images. *Future Generation Computer Systems*, *90*, 86-93.
19. Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). “Imagenet classification with deep convolutional neural networks”, *Advances in Neural Information Processing Systems*, 1097–1105.
20. Książek, Wojciech, Moloud Abdar, U. Rajendra Acharya, and Paweł Pławiak. "A novel machine learning approach for early detection of hepatocellular carcinoma patients." *Cognitive Systems Research* 54 (2019): 116-127.
21. Lakhani, Paras, and Baskaran Sundaram. (2017). "Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks", *Radiology*, 284(2), 574-582.
22. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. and Jackel, L. D. (1989). "Backpropagation applied to handwritten zip code recognition", *Neural Computation*, 1(4), 541–551.
23. Lee, J. G., Jun, S., Cho, Y. W., Lee, H., Kim, G. B., Seo, J. B., & Kim, N. (2017). Deep learning in medical imaging: general overview. *Korean journal of radiology*, *18*(4), 570-584.
24. Liao, Haofu. "A deep learning approach to universal skin disease classification." *University of Rochester Department of Computer Science, CSC*(2016).
25. Lopez, Adria Romero, Xavier Giro-i-Nieto, Jack Burdick, and Oge Marques. "Skin lesion classification from dermoscopic images using deep learning techniques." In *2017 13th IASTED International Conference on Biomedical Engineering (BioMed)*, pp. 49-54. IEEE, 2017.
26. Mostafa, Salama A., Aida Mustapha, Mazin Abed Mohammed, Raed Ibraheem Hamed, N. Arunkumar, Mohd Khanapi Abd Ghani, Mustafa Musa Jaber, and Shihab Hamad Khaleefah. "Examining multiple feature evaluation and classification methods for improving the diagnosis of Parkinson’s disease." *Cognitive Systems Research* 54 (2019): 90-99.
27. Mukherjee, Soumen, Arunabha Adhikari, and Madhusudan Roy. (2019). "Melanoma Identification Using MLP With Parameter Selected by Metaheuristic Algorithms", *Intelligent Innovations in Multimedia Data Engineering and Management. IGI Global*, 241-268.
28. Ronneberger, O., Fischer, P. and Brox, T. (2015). “U-net: Convolutional networks for biomedical image segmentation”, *in International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 234–241.
29. Roth, H. R., Lu, L., Seff, A., Cherry, K. M., Hoffman, J., Wang, S., Liu, J., Turkbey, E. and Summers, R. M. (2014). “A new 2.5 d representation for lymph node detection using random sets of deep convolutional neural network observations”, *in MICCAI 2014. Springer*, 520–527.
30. Setio, A. A. A., Ciompi, F., Litjens, G., Gerke, P., Jacobs, C., van Riel, S. J., Wille, M. M. W., Naqibullah, M., S´anchez, C. I. and van Ginneken, B. (2016). “Pulmonary nodule detection in ct images: false positive reduction using multi-view convolutional networks”, *IEEE transactions on medical imaging*, 35(5), 1160–1169.
31. Shin, H.C., Roth, H. R., Gao, M. , Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D. and Summers, R. M. (2016). “Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning”, *IEEE transactions on medical imaging*, 35(5), 1285–1298.
32. Shoieb, Doaa A., Sherin M. Youssef, and Walid M. Aly. (2016). "Computer-aided model for skin diagnosis using deep learning", *Journal of Image and Graphics*, 4(2), 122-129.
33. Simonyan, Karen, and Andrew Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv*, 1409-1556.
34. Tajbakhsh, N. , Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B. and Liang, J. (2016). “Convolutional neural networks for medical image analysis: Full training or fine tuning?”, *IEEE transactions on medical imaging*, 35(5), 1299–1312.

.