**BACHELOR OF SCIENCE IN ELECTRONIC AND TELECOMMUNICATIONS ENGINEERING**

# Detection of Skin Cancer Using Deep Learning Approach


**Submitted By:**

**Minul Hasan Miner**
**T-183036**


**Submitted To:**


**Engr. Md. Razu Ahmed**
**Professor & Chairman**
**Department of CCE**
**International Islamic University Chittagong**

**Department of Electronic and Telecommunication Engineering**
**International Islamic University Chittagong**
**Kumira, Sitakunda, Chattogram**


**July, 2023**

## DEDICATION

This thesis is dedicated to all of our honourable parents and teachers.

## CERTIFICATE OF APPROVAL

The Thesis entitled as "**Detection of Skin Cancer Using Deep Learning Approach**" Submitted by Minul Hasan Miner & ID No: T183036, to the Department of Electronic and Telecommunications Engineering (ETE) of International Islamic University Chittagong (IIUC) has been accepted as satisfactory for the partial fulfillment of the requirements for the Degree of Bachelor in Electronic and Telecommunications Engineering and approved as to its style and contents for the examination held on____________2023.

Approved by

**Engr. Md. Razu Ahmed**

Supervisor

Professor & Chairman
Department of CCE
International Islamic University Chittagong
Kumira, Sitakunda, Chittagong

# CANDIDATES DECLARATION

It is hereby declared that the work presented in this thesis has not been submitted elsewhere for the award of any degree or diploma, does not contain any unlawful statement.

________________

**Name:** Minul Hasan Miner

**Id:** T183036

## SUPERVISOR’S DECLARATION

I therefore testify that we have read this thesis and that, in terms of scope and quality, it satisfies the requirements for the conferring of a Bachelor of Science in Electronic and Telecommunication Engineering (ETE).

Saturday, $15^{th}$ July, 2023

**Engr. Md. Razu Ahmed**

Professor & Chairman
Department of CCE, IIUC

**Thesis/Project Report Declaration and Copyright Thesis/Project Report Title:**

**AUTHORS:**

| SL NO. | AUTHOR’S NAME | STUDENT ID | SIGNATURE |
|---|---|---|---|
| **1** | MINUL HASAN MINER | T183036 | |
| **NAME OF SUPERVISOR:** ENGR. MD. RAZU AHMED<br><br>**SIGNATURE OF SUPERVISOR:** | | | |

**We claim,**

1. Our thesis will be made available as an open access (full text) online database or repository at IIUC.

2. IIUC's Department of ETE retains the following rights:

   i. The Department of Electronic and Telecommunications Engineering (ETE) at IIUC is the inspiration of the thesis.

   ii. The IIUC Library has the authority to make copies for research purposes exclusively.

   iii. The thesis has the right to be copied by the Library for scholarly purposes.

# ACKNOWLEDGEMENTS

First and foremost, we would want to express our heartiest gratitude and humble submission to the one and only almighty Allah. Without Allah's goodwill and guidance, we would not be able to complete our task within the stipulated period of time. It was a terrific opportunity for us to study at the International Islamic University Chittagong's Department of Electronic and Telecommunication Engineering (ETE). We express our deepest gratitude to our department and professors. Without the assistance and support of a large number of other people, this thesis would not have been completed. We'd also like to thank Engr. Md. Razu Ahmed, Chairman & Professor, Department of CCE (IIUC) International Islamic University Chittagong, for his kind and helpful suggestions throughout the thesis process. His constant support and valuable advices helps us to decide the thesis topic. Under his proper supervision and guidelines we have been able to complete this thesis timely. We are also thankful to all of our honorable professors for their teaching, support, inspiration and guidance. Finally, we are conveying our whole gratitude to our parents and family who were always there to support us.

# ABSTRACT

Critical research is presently being done in the field of computer visions to classify and identify skin cancer. Several deep convolutional neural networks were used by researchers to enhance the performance of the current systems. There have been several efforts made in the past to identify skin cancer. To increase performance and accuracy, many researchers employ a variety of efficient procedures. In this thesis project, we are attempting to build a model for identifying skin cancer based on method (DenseNet-121). For training and testing purposes in detecting skin cancer, we employed a dataset. Our suggested model has a 92% accuracy rate.

# TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF ILLUSTRATIONS

## LIST OF ABBREVIATIONS

| | |
|---|---|
| IIUC | International Islamic University Chittagong |
| ResNet | Residual Network |
| DenseNet | Densely Connected Convolutional Networks |
| CNN | Convolutional Neural Network |
| RNN | Recurrent Neural Networks |
| SVM | Support Vector Machine |
| ANN | Artificial Neural Network |
| NLP | Natural Language Processing |
| LSTM | Long Short Term Memory |
| ILSVRC | ImageNet Large Scale Visual Recognition Challenge |
| AK | Actinic Keratoses |
| BCC | Basal Cell Carcinoma |

# CHAPTER I

# INTRODUCTION

## 1.1 RESEARCH BACKGROUND

With a rate of fatalities of one out of every six people, cancer is one of the main causes of death in the modern era. One of the most deadly malignancy in the US is skin cancer. Integrating traditional machine learning procedures like Support vector Machine, Random forest, etc. with manual feature extraction techniques has shown effective outcomes in the diagnosis of melanoma [7, 8]. A carefully designed feature extractor that can represent data features that are acceptable for classifier learning is necessary because typical ML algorithms are restricted in their ability to analyses data in its raw form skin cancer signs are-

1. A fresh skin lesion or.

2. A previous region that experienced a size, shape, or color change.

3. A region that is difficult or irritating.

4. An incision that bleeds or produces a crust and won't heal.

5. A glossy, skin-colored or red bump on the skin's surface.

6. A palpable red spot that is rough or scaley.

Deep learning has evolved into a subset of the field of machine learning (ML) in order to address the difficulties which regular methods for ML confront. The representation learning technique provided by deep learning made it possible to gather features for detection, classification, or additional uses just by supplying raw data.

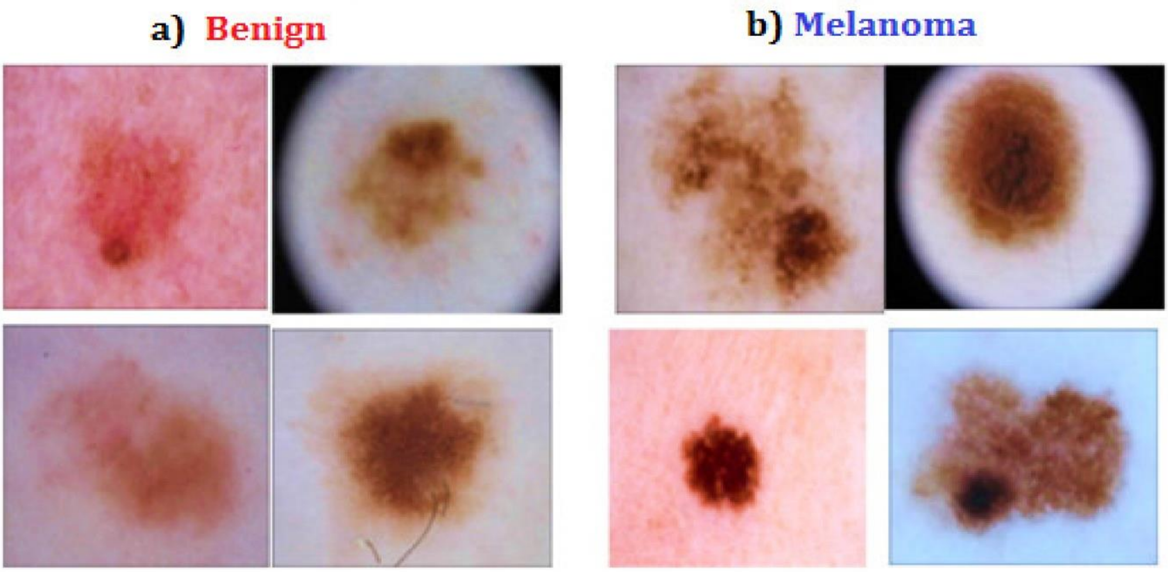


Figure 1.1: Skin cancer

### 1.1.1 Image Processing

Image processing is a technique for enhancing or extracting information from a picture by applying operations to it [17]. It's a type of signal analysis in which an image is the input and an image or the image's features/characteristics are the output. However, it must process photos in an additional phase after they have been captured in order to set up an ideal workflow and avoid time loss.

The following are the three processes in image processing:

1. Importing a picture utilizing image-capture software.

2. Analyzing and editing the image.

3. An improved image or a report based on the image analysis can be the output.

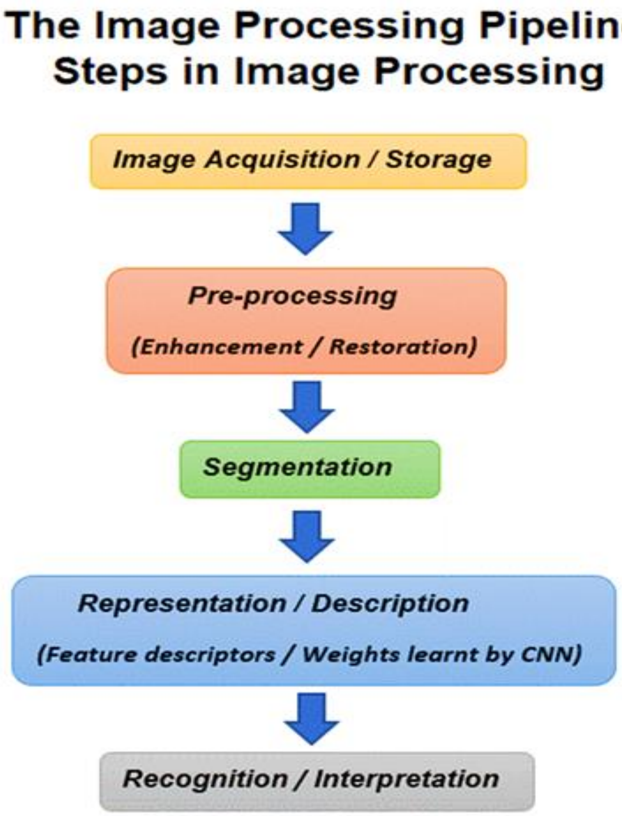


Figure 1.2: Image Processing

### 1.1.2 Skin Cancer

Skin cancer, or the abnormal the addition of skin cells, is most frequently found on exposed skin to the sun. However, even on parts of your skin that are seldom exposed to sunlight, this prevalent kind of cancer may show itself. The three primary kinds of skin cancer are basal cell carcinoma, squamous cell carcinoma, and melanoma.

### 1.1.3 Types of skin cancer

According to dermatology, there are 4 kinds of skin cancer –

1. Basal cell carcinoma

2. Squamous cell carcinoma

3. Merkel cell cancer

4. Melanoma

## 1.2 PROBLEM STATEMENT

Skin cancer are becoming an increasingly widespread problem in modern society. The majority of these disorders are serious and harmful, especially if they are not treated early on. Skin problems are not taken seriously by the general public. The majority of people manage skin infections with their own home remedies. However, if these home remedies aren't appropriate for that particular skin problem, the skin will be damaged. They may also be unaware of the seriousness of skin problems. Skin infections have a high tendency for spreading from one individual to another. As a result, it's critical to keep it under control at an early stage in order to prevent it from spreading to humans. Damage to the skin caused by skin diseases has the potential to harm people's self-confidence, mental health, and overall well-being. As a result, skin cancer have become a major public health issue. It has become critical to treat these skin cancer correctly at an early stage in order to avoid major skin damage. This system would go a long way toward solving the problem. Because the system would allow users to diagnose skin cancer and provide remedies or advice to patients using photos of cancer affected skin and information from the patient, it would be useful.

## 1.3 MOTIVATION

• Skin cancer is a significant current problem. People don't understand the harmful aspects of skin cancer and don't give much importance to skin cancer until it reaches the extreme level. If we can successfully detect cancer then many problems will be solved. People will be able to identify their own skin diseases in a short time. They will understand how harmful their skin disease is and how contagious it is.

• These skin disorders may worsen over time if they are not discovered and identified at an early stage. It could lead to the spread of the disease and worsening of the situation.

They can spread from one portion of the body to another, or from one person to another. In these situations, the best bet is to spot them as soon as possible. Because the characteristics of skin disorders are so different, developing a system or algorithm that can reliably and efficiently detect them becomes a difficult task. Because skin cancer vary from person to person, the color of the skin, as well as the color of the skin itself, is significant in detecting them.

• If we can do this, we will be able to research more in the future.

• If we can detect skin cancer image properly and accuracy is good, then we will try to implement more model later.

### 1.4 OBJECTIVE OF RESEARCH

The goal of this study is to come up with innovative techniques. ResNet50 for skin diseases detection.

1. To generate a novel densenet121 approach to enhance performance on the little dataset.

2. Utilizing training samples of multiple images per class, the CNN classifier system is able to recognize skin cancer from cancer images.

3. To detect patient's own skin cancer in a cost-free manner.

### 1.5 ORGANIZATION OF THE THESIS

The thesis is divided into five chapters. The structure of the thesis is described in general terms below:

1. The thesis beginning is presented in the first chapter. This chapter describes the problem and the motivation for the research. The research objectives of the thesis are also outlined.

2. The second chapter provides an overview of the Literature Review and Research Scope. There also discuss about those papers research gap and how overcomes those short comes.

3. The third chapter describes Methodology, Research activity. We have presented data processing, data collection, preparing the model, training the model, optimizer.

4. Results and discussion are presented in the fourth chapter. We have discussed the performances of the model.

5. The fifth chapter discusses the thesis's conclusion, contribution, and future works.

# CHAPTER II

# LITERATURE REVIEW

## 2.1 INTRODUCTION

A skin cancer detection and classification system detect whether cancer is present and, if it is, classifies the type of disease that is there [10]. The classification is based on decisions made using feature extraction methods to extract features. Previous research has shown that a lot of work has been done on skin cancer. There are various algorithms of machine learning such as CNN, RNN, ANN, CNN-SVM, Random forest have been used. But not much has been done with DenseNet-121. So here we have decided to use the DenseNet-121 algorithm for this purpose.

## 2.2 SCOPE OF RESEARCH

The scope of a study describes the scope of the task and the parameters that will be employed to conduct it. To put it another way, you'll need to figure out what the research will cover and what it will focus on.

There have many shortcomings found in this research field. Notable among the previously used machine learning algorithms for skin cancer detection are CNN, RNN, ANN. However, the use of CNN is the most, but not much work with DenseNet-121. We need to see how well DenseNet-121works. So we tried to participate in the field to see the performance of DenseNet-121 in skin cancer detection.

## 2.3 LITERATURE REVIEW

Several researchers have proposed image processing-based methods for detecting different types of skin disorders. We'll go over some of the strategies that have been documented in the literature.

### 2.3.1 RNN

An artificial neural network (ANN) called a recurrent neural network processes data in a sequential or time-series fashion. Deep learning algorithms are applied in popular apps like Siri, voice search, and Google Translate for ordinal or temporal difficulties involving language translation, natural language processing (NLP), speech recognition, and picture captioning. Convolutional neural networks (CNNs) and feedforward neural networks are an instance of recurrent neural networks that learn from training input. Their "memory", that allows them to alter current input and output using information from prior inputs, sets them apart from various other systems. Recurrent neural network's output depends on earlier segments of the sequence, although the inputs and outputs of traditional deep neural networks have been engineered to be distinct from one another. Unidirectional recurrent neural networks are unable to include future events into its forecasts, despite having the potential that they would affect the eventual result of a series. Different RNN methods for designing.

**Bidirectional Recurrent Neural Networks (BRNN):**

These RNNs have a particular network construction. Bidirectional RNNs may use future data for better accuracy compared with unidirectional RNNs, which simply generate predictions about the current state based on recent inputs.

**Long Short-Term Memory (LSTM):**

This well-known RNN architecture was devised by Sepp Hochreiter and Juergen Schmidhuber to solve the vanishing gradient problem. They are trying to find resolution to the problems caused by sustained reliance. If the prior state that affected the current forecast was not recent, the RNN model might not be able to reliably forecast the current state. Assume if we wanted to be prepared for the words in italics "Alice is allergic to nuts." She can't consume peanut butter since she has aversions to it." We can predict that the meal we can't consume contains nuts when we examine the setting of a nut allergy. However, the RNN would have trouble, if not fail, to connect the data if the context existed a few words earlier. This is taken care of by LSTMs, which make use of "cells" in the neural network's hidden layers, which comprise of three gates: an input gate, an output gate, and a forget gate. The data flow essential for forecasting the network's output is controlled by these gates. If gender pronouns, such as "she", were used numerous times in preceding sentences, you can leave them out of the cell state.

**Gated Recurrent Units (GRUs):**

In RNN models, this RNN variant is comparable to LSTMs in that it solves the short-term memory problem. It governs data using hidden states rather than "cell states", and instead of three gates, it only has two: an update gate and a reset gate. Like the gates in LSTMs, the reset and update gates govern how much and which information is preserved.

**Recurrent Neural Network Benefits (RNN):**

An RNN evolves to remember every item from data it encounters over time. It can only be employed in time series prediction because it can remember past inputs. This is commonly known as long short term memory (LSTM).

With recurrent neural networks, even convolutional layers are used to expand the effective pixel neighborhood.

### 2.3.2 CNN

One or more convolutional layers are included in a CNN, and each of them are followed by one or more fully linked layers. We may include particular traits in the design because CNN's considers that the inputs are images [20]. This leads to a superior function because there are simply fewer properties in the network. Another advantage that convolutional neural networks have over fully interconnected networks is that they are easier to train. There are four basic actions that can be seen in convolutional networks.

1. Convolution is number one

2. Non-linearity is the second factor to consider

3. Creating a pool (Subsampling)

4. The classification system (Fully-connected)

A typical digital camera image comprises channels that are red, green, and blue. Pixel values range from 0 to 255 in these three-layered 2D matrices. A convolutional neural network (CNN) has layers which each transform an input layer into a differentiable output layer. Fully-connected, convolutional, and pooling layers are the three various kinds of layers that compose convolutional neural networks (CNN). Convolutions are a great way for assigning a layer depth. The convolutional layer is the most essential component of a convolutional network. This layer handles the majority of the computational heavy lifting. Learnable filters, often known as kernels, are used to create the parameters of convolutional layers [15]. Every filter is made up of a receptive field, which is a small portion of an image. The width of the filter is comparable to the input volume's apply depth. The filter on a neural network layer is an array of numbers with each of the following dimensions: w pixels wide, d pixels deeply, h pixels high, and the color channels. The filter multiplies its values by the input image's pixel values as it slides or convolves along the width and height of the image. When pushed over the input volume, each convolutional layer contains a collection of filters that generate a 2-dimensional activation map. The output volume is increased by stacking them along with the depth. A hyperparameter that describes the geographic scope of this connectivity is the neuron's receptive field. Each neuron in the convolutional layer will have weights to a region in the input volume that has a total of x*y*d weights if the input volume utilized by this study is w x h x d and the receptive size of the field is x x y. The three parameters depth, stride, and zero-padding equally impact the output volume.

• The depth of learning has an inverse correlation with the number of filters used in searching for anomalies in the input. Different neurons along the depth dimension are triggered when the raw image input to the first convolutional layer is used, despite the fact there are differentially oriented edges, colors, and so on.

• The filter slides through the input volume to generate stages. The stride is the distinction in the filter's placement. Consequently, the stride is 1 when the filter is moved one pixel at a time. The stride is two when the filter hops two pixels at a time as it circles the input.

• A hyperparameter determines the size of the zero padding. The spatial size of the output volume decreases as filters are applied to an input volume. When more convolutional layers are added, the output volume decreases. When extracting low-level qualities from picture borders, it is sometimes valuable to keep as much data from the original input volume as you can. As a result, the input volume and output volume are often kept at the same spatial size. You can alter the output volume's spatial size by using the zero padding feature. It accomplishes this by padding the input volume entirely with zeros [22].

The output volume's spatial dimension can be determined by calculating the input size (W), the convolutional layer neurons' receptive field size (F), the strides utilized (S), and the zero padding used on the border (P).

The spatial size of the output volume is obtained as $(W-F+2P)/S+1$.

**Pooling layer:**

In a neural network, a pooling layer is often positioned between subsequent convolutional layers. In order to lessen the constraints on the network, it lowers the spatial size of the representation. In contrast to averaging or other methods, a pooling layer operates independently on each depth slice of the input and partially reduces it using the MAX operation [11]. It is accepted to use a 2x2 filter and a stride of the same length, with a maximum of four digits needed for each MAX operation. The following are some pooling layer qualities in general.

### 2.3.3 VGG16

A convolutional neural network (CNN) named VGG16 won the 2014 ILSVR (ImageNet) competition. It is frequently considered as one of the most comprehensive vision model designs ever produced. The most noteworthy feature of VGG16 is that it focuses on having 3x3 filter convolution layers with a stride 1 consistently employs the same padding and maxpool layer of 2x2 filter stride 2. The max pool and convolution layers are always paired together in the architecture. The output is finished by two FC (completely connected layers) and a softmax. The fact that it contains 16 layers with various weights is where the term "VGG16" originates. This network is quite big, with 138 million parameters, based on estimates [21].

VGG16 is an object identification and classification system that can classify 1000 images into 1000 different categories with a precision rate of 92.7%. It is a widely used technique for identifying images that is relatively easy to use to transfer learning. Because of its advantages, VGG16 is often utilized in learning applications.

**Challenges of the VGG 16**

• It takes a long time to train (on the Nvidia Titan GPU, the original VGG model needed 2-3 weeks).

• Trained picture with VGG-16528 MB are taken up by net weights. It's ineffective because it uses a lot of bandwidth and storage space as a result.

### 2.3.4 ResNet50

ResNet-50 is a 50-layer deep CNN that has forty-eight convolution layers, one average pool layer, and one max pool layer. Floating-point operations are included. Because ResNet50 is a popular ResNet model, we've gone through it in depth. You can import a network that has already been trained using more than a million images from the ImageNet database. The network has a capacity to identify images into 1000 different object categories, such as keyboards, mice, pens, and other animals. The network has amassed a library of in-depth feature representations for a variety of images as a result. The network's picture input size is 224×224 pixels [6].

This architecture can be used to computer vision tasks including object localisation, object recognition, and picture classification. It can also be extended to non-computer vision applications to provide them the benefit of depth while minimizing computational costs.

With ResNet Networks, you can train a lot of layers up to thousands without increasing the training error%. ResNets can help with the vanishing gradient issue by using identity mapping [8].

### 2.3.5 ImageNet

The ImageNet dataset has long been regarded as a watershed moment in computer vision. The scale and semantic diversity of ImageNet have had a significant impact on computer vision research.

A consortium of professors and researchers from Princeton, Stanford, and UNC Chapel Hill generated the ImageNet collection. The original purpose of ImageNet was to fill the WordNet hierarchy with around 500-1000 photos per concept. Images for each concept were acquired by querying search engines and submitting potential images to Amazon Mechanical Turk for evaluation. ImageNet has grown in popularity as a large-scale training corpus for computer vision as well as an evaluation benchmark throughout the years.

Developing in popularity is the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), which uses some of the ImageNet dataset with 1000 item classes for testing picture classification systems.

**ImageNet Applications**

• Pre-training

Until recently, the fundamental backbone for pre-training in computer vision was given by projects like CLIP, ImageNet, and COCO. The deep learning system learns to extract features from photos that are significant in general during pre-training. The pretrained network is then focused into a specific domain of interest through finetuning using a custom dataset.

• Benchmarking

ImageNet has laid a good framework for measuring progress in computer vision research, notably in image classification. The vast ImageNet dataset is made up of tagged images that have been gathered by researchers to help in the creation of computer vision algorithms. An annual contest called the ImageNet Large Scale Visual Recognition Challenge, or ILSVRC, promotes the creation and evaluation of cutting-edge algorithms using subsets of the ImageNet dataset. The ILSVRC requirements have led to a variety of model architectures and approaches at the interface of deep learning and computer vision.

**DenseNeT-121**

In an ordinary feed-forward convolutional neural network (CNN), only the first convolutional layer, which receives the input, receives the output of the convolutional layer preceding it. The output feature map from this convolutional layer is subsequently sent to the following convolutional layer. There are L direct connections for 'L' layers, one between each layer and the one after it. However, if the CNN has more layers or becomes deeper, the 'vanishing gradient' problem starts to show itself. This means that when the communication channel between the input and output layers gets longer, some information may "vanish" or "get lost," which reduces the network's ability to train properly.

DenseNets resolve this problem by tweaking the usual CNN architecture and streamlined the connectivity between layers. The term "Densely Connected Convolutional Network" refers to an architecture in which every layer is directly connected to every other layer. Between 'L' layers, there are L(L+1)/2 direct connections.

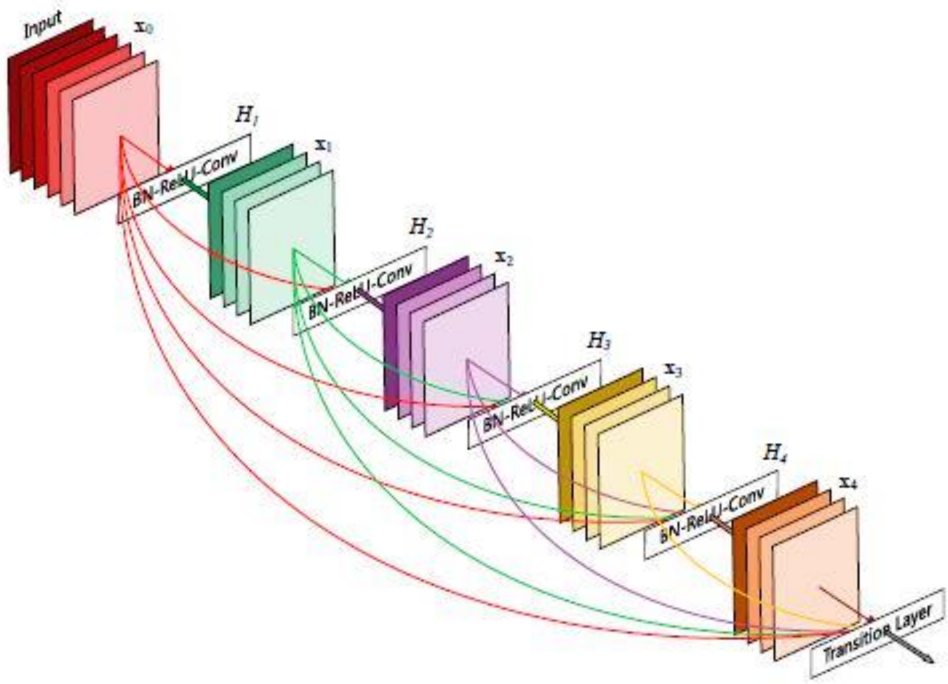


Fig 2.1: DenseNet-121

In short, DenseNet-121 has the following layers:

1. 1 7x7 Convolution
2. 58 3x3 Convolution

3. 61 1x1 Convolution
4. 4 AvgPool
5. 1 Fully Connected Layer

### 2.3.6 Existing work on skin disease

**General**

In [2] One of the dangerous cancers is skin cancer. In essence, it is a unique growth of skin cells that primarily appears after skin contact with the sun. These days, it can even appear on skin places that are not exposed to sunlight. Cancer must be detected early if it is to avoid slowly spreading to other sections of the body. Here, we describe a variety of machine learning methods for identifying different cancer kinds and categorizing them as malignant or benign. Preprocessing, Segmentation, Feature Extraction, and Classification are the process's four primary steps. The comparison of several sorts of procedures that can be utilized for the aforementioned processes is the main focus of the paper. They proposed the Multi-Channel-ResNet architecture for diagnosing medical pictures. They used two types of data to verify the system: dermoscopic pictures and upper-skin surface images. They gathered surface photos with four illnesses, however the framework only obtained an accuracy of 82.4%, which is reasonably acceptable but not adequate for identifying diseases smoothly [1].

In [3] The most prevalent type of cancer is by far skin cancer. If discovered quickly, this can be efficiently treated. Early cancer detection is quite expensive. The abnormal development of skin cells is known as skin cancer. If caught early and addressed, these are highly curable. Actinic keratoses (AK), basal cell carcinoma (BCC), dermatofibroma, and melanoma are the four different kinds of skin cancer. A late diagnosis of cancer causes the disease to spread to other organs. Using convolution neural networks, the skin cancer may be identified from the photos. This implementation will make use of the HAM10000 and ISIC image datasets. The model in CNN's performs better as a result of transfer learning.

To extract characteristics from pre-trained models, which are then used to categorize different forms of skin cancer, go here. Some of the machine learning and deep learning techniques utilized in this implementation are Random Forest, SVM, CNN, and DenseNet [3]

In [4] Skin cancer is a serious condition that affects practically all age groups, but it is more common as people get older. The epidermal layer of the skin's growth is aberrant or fast, which promotes the development of tumors. Cancer needs to be distinguished from other skin conditions because it affects tissue levels, whereas other conditions simply affect the surface layer of epidemics. Technology can aid in early cancer detection, which lowers the cost of cancer detection while also saving time because cancer detection and treatment are both expensive.

## 2.4 SUMMARY

We discussed many papers which are related to our thesis work. In this chapter, some paper's review we have added. The works which are previously done showed that DenseNet-121 is often used for detection of skin cancer and results are good.

# CHAPTER III

# METHODOLOGY

## 3.1 INTRODUCTION

Our research methodology will be discussed in this chapter. Skin cancer detection is established using Deep Learning in this study. There will be a discussion about the process of collecting data and explaining the data. We will then discuss the proposed model. Finally, the chapter finishes up with a look at the Parameter Optimizer and the Feature Selection method.

## 3.2 METHODOLOGY

In computer vision, detecting skin diseases from a color image is a difficult issue. The major goal of our research is to use the DenseNet-121 model to detect skin diseases. There are also tests on how to get the most accuracy.

**Stage 1 – Data collection:**

We acquired data on the web and constructed our data collection via raw data processing.

There are many places to collect free data. Some of these are -

- Google Dataset Search.
- Data govt.
- Datahub.io.
- Earth data.

• Global Health Observatory data Repository.

We have collected our raw data from Kaggle, Google dataset, Data.Gov.

**Stage 2 - Data processing:**

All data gathered from different sources have been gradually processed. There is a lot of noise and inaccurate data. We process the data first and then use the selected data set in the following phase.

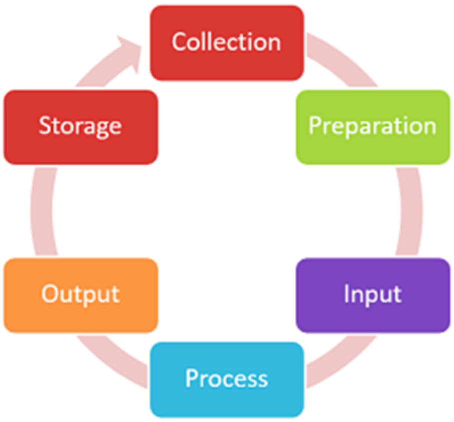


Figure 3.1 : Data Processing

**Stage 3 – Resize and increase:**

Knowing how to accurately resize an image is essential because most photos aren't the exact size we needed. The pixels in an image are updated as it is resized. There was an increase in accuracy when the image size was larger than the suggested architecture size, followed by a steady loss in accuracy .

The dataset was less so the accuracy didn't come much, later we increased the dataset to increase the accuracy.

**Stage 4 – Selection of model:**

There's many types of machine learning algorithms for recognizing skin cancer. Logistic regression, kernel SVM, Nave prejudice, random forest and CNN, VGG 16, ResNet 50, and AlexNet are notable algorithms. In this study, the DenseNet 121 algorithm was implemented to find conditions of the skin.

In an ordinary feed-forward convolutional neural network (CNN), only the first convolutional layer, which receives the input, receives the output of the convolutional layer surrounding it. The output feature map from this convolutional layer is eventually sent to another convolutional layer. There are L direct connections for 'L' layers, one between each layer and the one after it.

However, if the CNN has more layers or becomes deeper, the 'vanishing gradient' problem tends displaying itself. This means that when the transmission channel between the input and output layers gets longer, some information may "vanish" or "get lost," which reduces the network's ability to train properly.

DenseNets resolve this problem by tweaking the usual CNN architecture and facilitating the connectivity between layers. The term "Densely Connected Convolutional Network" refers to an architecture in which every layer is directly connected to every other layer. Between 'L' layers, there are L(L+1)/2 direct connections.

**Stage 5 – Evaluation of performance:**

The performance here was not taken at once. When coding we first use some datasets from Kaggle. Then we check the accuracy. Next, we collect many types of raw data from Google and expand the dataset. For which accuracy increases further. We have also made a lot of changes in the line of code to improve the performance.

**Stage 6 –Workflow:**

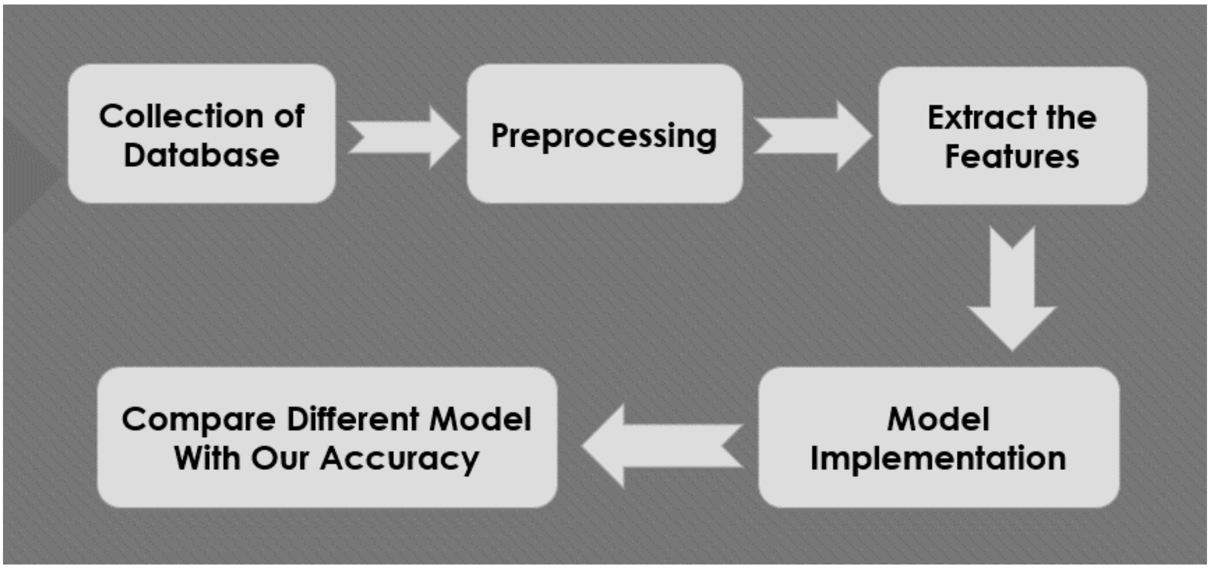


Figure 3.2: Workflow

## 3.3 DATA

### 3.3.1 Data collection

We collected our data from the web and constructed our data collection via raw data processing.

There are many places to collect free data. Some of these are

- Google Dataset Search.
- Data.Gov.
- Datahub.io.
- CERN Open Data Portal.

- Data repository used by the Global Health Observatory.

- UCI Machine Learning Repository.

- Earth Data.

We have collected our raw data from Kaggle, Google dataset, Data.Gov, Adobe stock, DermNet NZ. In 100% of the data, we tested 80% and trained 20%. We have used five classes of images in skin cancer detection.

#### 3.3.2 Data processing

Data processing is the process of shifting data from one form to one that is substantially more useful and desired, i.e., making it more pertinent and beneficial. This process can be carried out with machine learning algorithms, mathematical modeling, and statistical data. Depending on the task at hand and the machine's requirements, the entire process can result in graphs, movies, charts, tables, photographs, and a variety of other formats. This is the stage where algorithms and machine learning approaches are required to accurately and efficiently execute the commands given over a large volume of data.

#### 3.3.3 Data augmentation

The term "data augmentation" in data analysis refers to methods for expanding the amount of data available by adding significantly altered copies of already existing data or creating new synthetic data from previously collected data. It serves as a regularizer during the training phase of a machine learning model and reduces over fitting.

**a) Random Rotations:** The use of image rotation to render the model invariant to the object's orientation is a widespread augmentation approach. Rotating images randomly can be possible in a range of 0 to 360 degrees but rotation range argument in the ImageDataGenerator class should be an integer number.

**b) Random Zoom:** For training our deep learning model, random zoom image augmentation is used to generate images with variable zoom levels. We'll use the Keras ImageDataGenerator class and give the zoom_range option to it.

**c) Horizontal and Vertical Shift Augmentation:** When you move a photograph, all of the pixels move in the same direction—for example, laterally or vertically—while the image's proportions stay the same. As a result, some pixels in the image will be cut away and in some of the images, new pixel values will be required. The width_shift range and height_shift range options in the ImageDataGenerator constructor control the volume of lateral and vertical shift, respectfully.

**d) Shear:** Shear is a transformation that skews the image. Consider moving the image's left edge up while moving the right edge down (or vice versa). Shear_range in degrees can be used to create a random rotation.

**e) Image Flips:** Setting horizontal_flip=True or vertical_flip=True will flip a picture horizontally or vertically. A flip has a 0.5 chance of happening. Keep in mind that vertical flips aren't always beneficial, depending on the objective. For example, if we wish to identify items in images, we usually don't see them upside down.

### 3.3.4 Data Preparation

The practice of purging and altering raw data before processing and analysis is known as data preparation. It is important to note that before recycling, routine tasks such as reformatting data, making data updates, and joining data sets to enrich data must be completed. It's essential to be able to resize photos before modeling. For instance, the test image we've been using has the dimensions (640, 360). We can resize it to (224, 224), in which case the width, in this case, is shrunk to 224 and the height is scaled to keep the aspect ratio of the image. When using the OpenCV function imread() to read an image file, the color order is BGR (blue, green, red). In Pillow, on the other hand, the color order is presumed to be RGB (red, green, blue).

We used both the Pillow function and the OpenCV function in our code, so we had to convert BGR and RGB.

### 3.4 PROPOSED CNN MODEL

In a typical feed-forward convolutional neural network (CNN), only the first convolutional layer, which receives the input, receives the output of the convolutional layer preceding it. The output feature map from this convolutional layer is subsequently sent to the following convolutional layer. There are L direct connections for 'L' layers, one between each layer and the one after it. However, if the CNN has more layers or becomes deeper, the 'vanishing gradient' problem starts to show itself. This means that

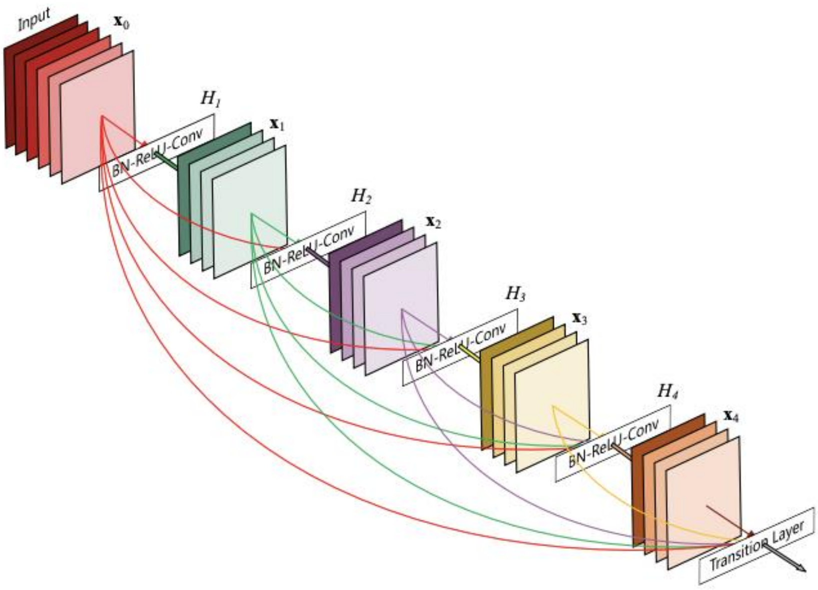


Figure 3.3 : Densenet121 basic architecture

when a communication channel across the input and output layers gets longer, some information may "vanish" or "get lost," which reduces the network's ability to train successfully. DenseNets resolve this problem by modifying the usual CNN architecture and simplifying the connectivity between layers. The term "Densely Connected Convolutional Network" refers to an architecture in which every layer is directly connected to every other layer. Between 'L' layers, there are L(L+1)/2 direct connections.

**Advantages:**

• Networks having a large number of layers, even thousands, may be trained simply without increasing the training error %.

• DenseNet121 can help with the issue of vanishing gradients by using identity mapping.

| Layers | Output Size | DenseNet-121 | DenseNet-169 | DenseNet-201 | DenseNet-264 |
|---|---|---|---|---|---|
| Convolution | $112 \times 112$ | $7 \times 7$ conv, stride 2 | | | |
| Pooling | $56 \times 56$ | $3 \times 3$ max pool, stride 2 | | | |
| Dense Block (1) | $56 \times 56$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$ |
| Transition Layer (1) | $56 \times 56$ | $1 \times 1$ conv | | | |
| | $28 \times 28$ | $2 \times 2$ average pool, stride 2 | | | |
| Dense Block (2) | $28 \times 28$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$ |
| Transition Layer (2) | $28 \times 28$ | $1 \times 1$ conv | | | |
| | $14 \times 14$ | $2 \times 2$ average pool, stride 2 | | | |
| Dense Block (3) | $14 \times 14$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$ |
| Transition Layer (3) | $14 \times 14$ | $1 \times 1$ conv | | | |
| | $7 \times 7$ | $2 \times 2$ average pool, stride 2 | | | |
| Dense Block (4) | $7 \times 7$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$ |
| Classification Layer | $1 \times 1$ | $7 \times 7$ global average pool | | | |
| | | 1000D fully-connected, softmax | | | |

Figure 3.4 : DenseNet121 model

## 3.5 PREPARING THE MODEL

### 3.5.1 Convolutional layers

A CNN is produced by the convolutional layer. Convolutional filters with a very small receptive field that cover the entire depth of the input volume comprise up the layer's parameters. Because it has a collection of filters, the parameters of a convolutional layer must be learned. The weight and height of the filter are both smaller than the input volume. For computing an activation map formed by neurons, each filter is convolved with the volume of input. Stacking the activation maps of every filter with the dimension of depth, convolutional layers output volume is acquired. Since each filter's height and width are designed smaller in size than the input, in the activation map every neuron is attached with a tiny local area of the input volume.

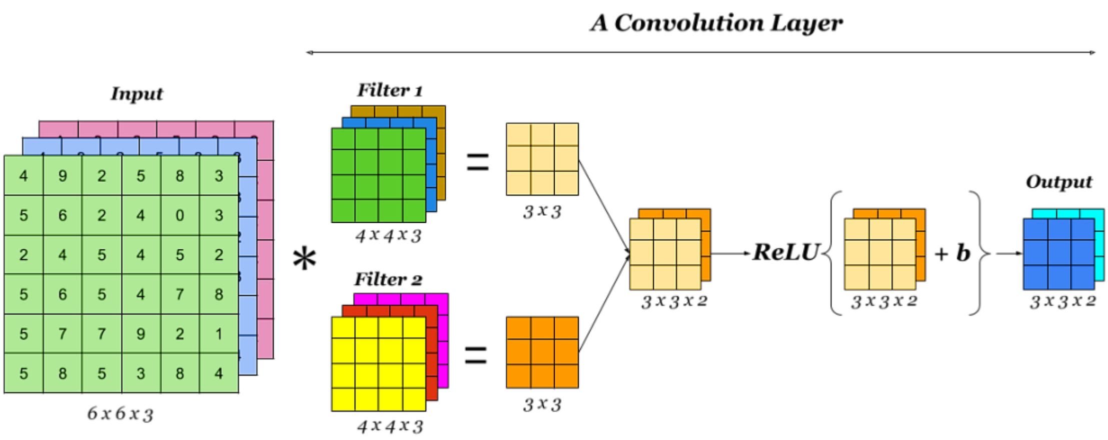


Figure 3.5 : Convolutional Layer

### 3.5.2 Feature extraction

Feature extraction is a dimensional reduction approach that breaks down a big quantity of raw data into manageable chunks. Because of the vast number of variables in these large amounts of data, a significant amount of computing power is required.

### 3.5.3 Activation function

Activation functions are computational operations connected to every neuron in a neural network which determine the network's output. The variables that guide whether a neuron should be activated or not are known as activation functions. The activation function is necessary in neural networks because it transforms the input in a nonlinear way and enables development.

• Binary Step Function:

It is a binary threshold oriented activation function. Here activation of neurons depends on the threshold value. Multi-value outputs are not permitted.

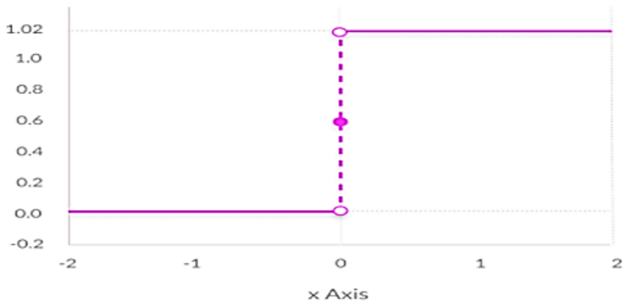


Figure 3.6 : Binary Step Function

• Function of Linear Activation:

The linear activation function can only be used in the output layer. It takes input and produce output proportional to input. Its equation is similar to straight line equation and range is – inf to + inf. It doesn't allow - backpropagation.

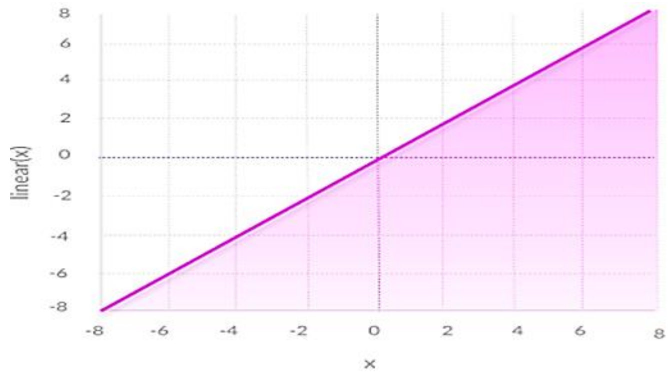


Figure 3.7: Linear Activation Function

• Functions of Non-Linear Activation:

Using these activation functions, the model can produce intricate mappings between the network's inputs and outputs, improving its capacity for learning as well as modeling.

Various types of non–linear functions are available.

Sigmoid or Logistic: This non-linear function was graphed as a ―S‖ shape. It is commonly used in the binary classification output layer, and its value range is 0 to 1. Its equation is:

A = 1/(1 + e-x) ……………………………………………….(2.1)

Its outputs are not zero centered and its computationally expensive.

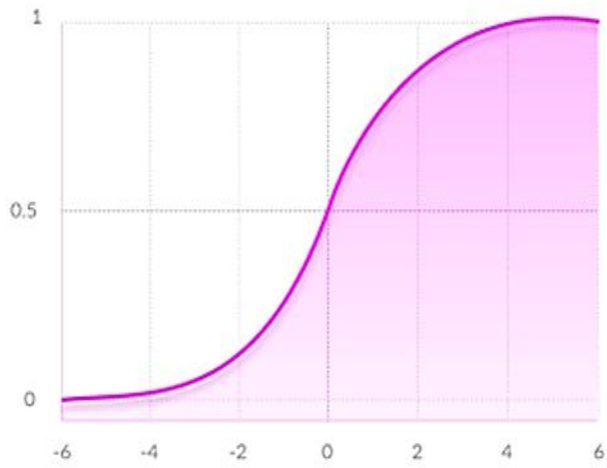


Figure 3.8 : Sigmoid Activation Function

• TanH:

TanH stands for hyperbolic tangent. TanH is also ―S‖ shaped . It is a more successful sigmoid function variation that has been mathematically relocated. Both function can be derived from each other. Its equation is

tanh (x)= 2 * sigmoid (2x) -1……………………………………(2.2)

• ReLu:

The abbreviation for ReLu is Rectified Linear Unit. It is mostly utilized in the network's hidden layers and is the most extensively used. Its computationally less expensive and efficient. It includes simpler mathematical operation and learns much faster than other activation function. Its equation is :

A(x) = max (0, x) ………………………………………………(2.3)

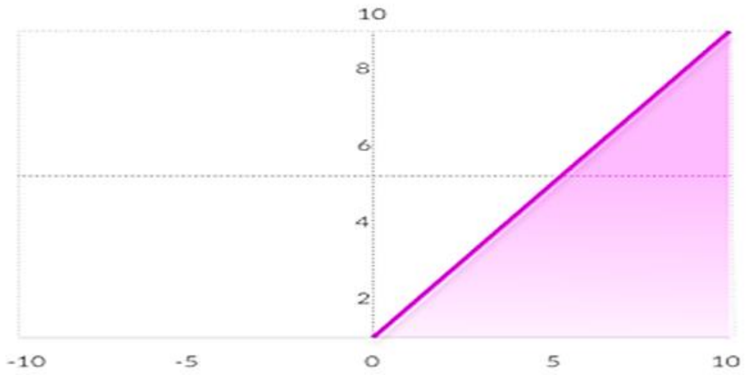


Figure 3.9: ReLu Activation Function

• Softmax:

Its kind a sigmoid function. Its handy for handling classification problem. It’s able to handle multiple classes.

### 3.5.4 Pooling layer

Another part of the CNN structure is layer pooling. Pooling layers are also used to minimize the spatial size of representations over time in order to reduce parameter calculation and network size. It operates independently on every map of feature. Inserting a pooling layer periodically into Convolutional Neural Network architecture within continuous Convolution layers is normal. Its goal is to reduce the amount of calculations and parameters in the network while also checking overfitting by serially shrinking the demonstration's geographical scale. The Pooling Layer works on each input's depth slice individually. The MAX technique is used to resize it.

• Max-Pooling:

Max pooling is a discretization approach based on samples. The goal is to reduce the dimensionality of an input representation (image, hidden-layer output matrix, etc.) so that characteristics in binned subregions can be assumed.

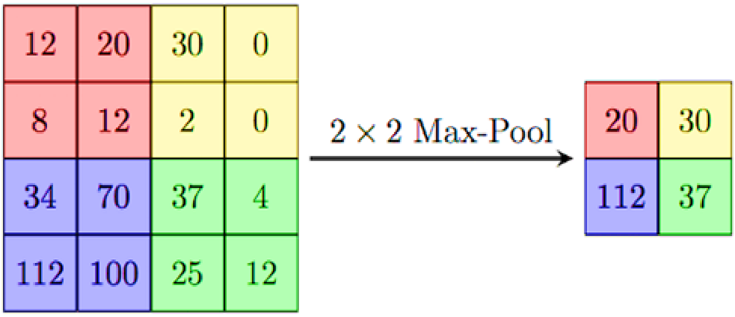


Figure 3.10 : Max-Pooling

• Average-Pooling:

Average pooling is the process of calculating the average for each region of the feature map.This implies that the average value is down sampled to each of the 22 squares on the feature map. A 66 feature map was the result of the line detector convolutional filter in the section before it.

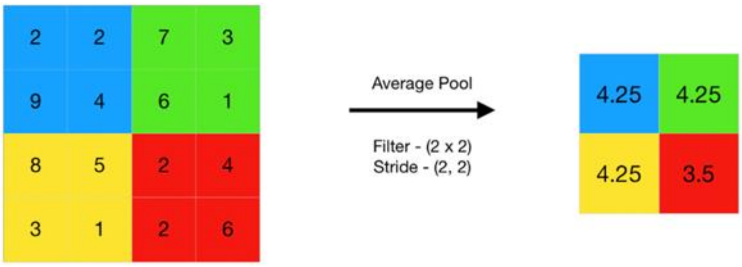


Figure 3.11 : Average-Pooling

### 3.5.5 Dense layer

Every neuron in the dense layer receives input from every other neuron in the existing layer, making it a deep-connected layer of the neural network. The dense layer has been shown to be the most often used layer in models. The dense layer combines matrices and vectors in the background. The matrix values, which are basically parameters, can be trained and altered using backpropagation. A dimensional vector is the dense layer's output. As a result, the dense layer is routinely used to alter the size of a vector. Dense layers also perform operations on the vector, such as rotation, scaling, and translation.

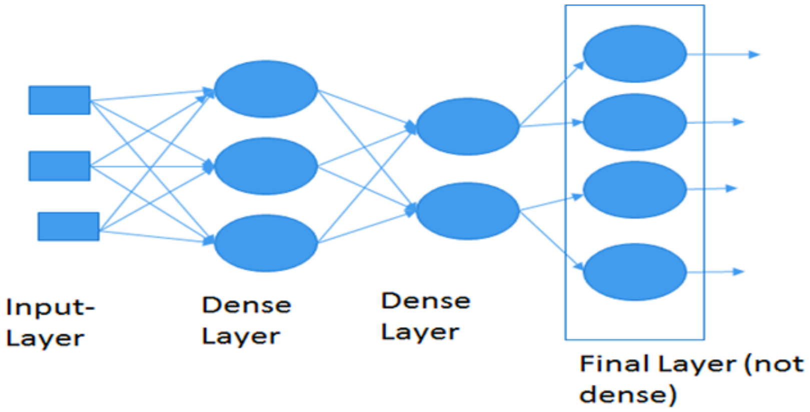


Figure 3.12: Dense Layer

### 3.5.6 Dropout

It's a regularization technique to prevent a model from being overfit. It randomly drop-out neurons from the network. It can be used after convolution layers and pooling layers. Its often used after pooling layers.

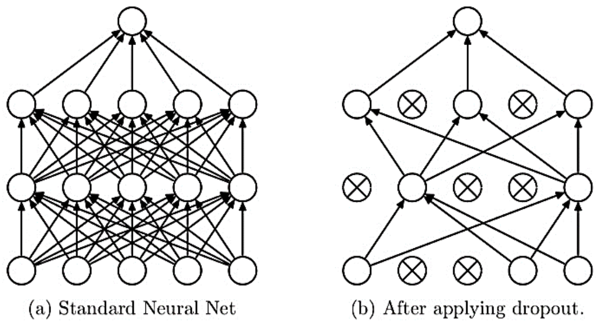


Figure 3.13 : Dropout

### 3.5.7 Flattern layer

A fully linked layer can be used to input a single column vector that is created from a two-dimensional feature matrix. Before sending the output of the convolution layer to the final classification layer for prediction, flatten it.

## 3.6 TRAINING THE MODEL

Learning (determining) strong values for all of the weights and bias from labeled data is all that training a model comprises. In supervised learning, a machine learning a model that minimizes loss; this process is referred to as empirical risk minimization.

One method for rapidly assessing an algorithm's performance on our assignment is to create a train and test split of our dataset. The model is set up and trained using a training dataset. Assume that the test dataset is entirely new and that the output values of the algorithm are concealed.

## 3.7 OPTIMIZER

We have used ADAM optimizer. Future deep learning applications in computer vision and natural language processing may make use of the extended stochastic gradient descent conduct known as the Adam optimizer. For image classification tasks, Adam is known to perform worse than SGD. The Adam optimizer is a well-liked adoptive learning rate optimizer for deep learning, particularly in computer vision. After a certain number of epochs, say 50 epochs, I've seen articles where the learning rate was decreased by dividing it by 10.

Adam is the top optimizer. If one wants to train the neural network more rapidly and efficiently, Adam is the optimizer to utilize. For sparse data, use optimizers with a dynamic learning rate. Min-batch gradient descent is the optimum option when using a gradient descent method.

## 3.8 CONCLUSION

We have used one dataset which is skin disease dataset for experiments. Our Proposed model is Densenet-121. An optimizer function is used which is ADAM. We used the Categorical cross-entropy implementations to run our proposed model on the skin cancer datasets.

# CHAPTER IV

# RESULTS AND DISCUSSION

## 4.1 INTRODUCTION

A thorough experimental inquiry into our suggested system can be found in this chapter. Here, we go over our model's implementation specifics. And explain the confusion matrix and other experimental results in detail. The below sections are organized as Datasets and experimental settings and Experimental tools and environments described in the section 4.2 as well as section 4.3 respectively. In the section 4.4 and 4.5 we described Implementation details and Experimental analysis. Discussion and Summary are added in section 4.6 and 4.7. Utilities have been experiencing tremendous reform in recent years all across the world.

## 4.2 DATASETS AND EXPERIMENTAL SETTINGS

We have collected our raw data from Google dataset, Data.Gov, Adobe stock, DermNet NZ. We have used these raw datasets to train and test our model. Our datasets have been divided into two sections. These are both training and testing. 80% of the photos are used to train the model. For testing, the remaining 20% of the dataset's photos have been used.

This research aimed at the improvement of new methods.

## 4.3 EXPERIMENTAL TOOLS AND ENVIRONMENT

In this system, different tools, packages, and environments were used for data preprocessing, training, and testing the models, and measuring their performance. This section will describe these tools and environments elaborately.

### 4.3.1 Programming Language

We have used the libraries of Python Programming language which are Tensorflow and Keras. Python is the most popular language used for machine learning as it has the largest collection of packages for implementing machine learning algorithms and working with data.

### 4.3.2 Integrated Development Environments (IDE)

**1.** In this experiment, we used the Tensorflow library, the Keras API, and the Python programming language. The Tensorflow library has an interface provided by Keras. It's a Python interface provided by an open-source library. Tensorflow, on the other hand, is a complete set of libraries and a flexible ecosystem of tools that allows developers to quickly design and deploy machine learning applications.

**2.** Google Colaboratory: You can write and run any Python code in the web using the unrestricted Google Research tool known as Colaboratory (or colab). It's mostly used for data mining, but it's also utilized for data analysis and machine learning. Colaboratory is a hosted Jupyter notebook that doesn't need to be installed and gives you free access to computer resources like GPU and TPU. Colab allows one to use and share jupyter notebooks with others without any kind of run and installation problems. It uses Google drive for saving files mainly, so one can load files from google drive and also from github. Some features of Google colab are:

• It specially worked with deep learning libraries.

• Available GPU’s in colab are Nvidia K80s, T4s and P100s.

• Can be shared easily without the problem of download, run and installation.

• The GPU can be used continuously 12 hours as back-end.

### 4.3.3 Libraries

• OpenCV :

Open Source Computer Vision is the abbreviation for a machine learning software library called OpenCV. OpenCV-Python is a NumPy-based Python bindings library for handling computer vision issues. OpenCV array formats are frequently translated into and out of NumPy arrays. For real-time image processing, OpenCV includes a plethora of built-in capabilities.

• Tensorflow:

A free and open-source machine learning and artificial intelligence software library is called TensorFlow. It's specifically designed for the operations needed in large deep learning models, and computation can be done on GPU, TPU, and other platforms.

• NumPy:

NumPy is a Python module that actually named as Numerical Transform and is mostly used for data manipulation with arrays. It was founded in the year 2005. NumPy is an open-source project that includes functions for working with linear algebra, metrics, and the Fourier transform.

• Matplotlib:

In Python, a comprehensive library called Matplotlib is used to create animated, static, interactive visualizations.

• Scikit-Learn:

It's also a free Python machine learning library with a variety of algorithms. It includes algorithms such as random forests, k-neighbors, and SVM. NumPy and SciPy are two Python Numerical and Scientific libraries that Scikit-Learn supports.

## 4.4 IMPLEMENTATION DETAILS

Using Tensorflow, Keras python library we evaluated the proposed model on the pictures of our datasets. For all the loss functions and optimizers it contained implemented methods we used.

• We used the ADAM Optimization and Categorical crossentropy implementations of Tensorflow library for running the model on the dataset.

• All the output category values from the Scikit-Learn library were one hot encoded.

• After the two convolutional layers and fully linked layers, dropout layers are added.

## 4.5 EXPERIMENTAL ANALYSIS AND COMPARISON

On that dataset, we operate the network, primarily justifying it with the validation set, and analyze accuracy using the test set. We quickly outline our training operation before delving into the model's performance specifics on several datasets.

In our simulations, we trained each system on a different dataset, and we made an effort to maintain the same architecture and hyperparameters across all of these different models. Five epochs are used in this thesis work for training. An empirical learning rate of 0.01 is used for optimization. This technique use the ADSM optimizer as the optimizer algorithm. The time involved to train a model on our datasets is less than reasonable.

In our investigation, the datasets were divided into two groups: training and testing. The training set, learning set, or training dataset refers to the body of knowledge that has been used to train the model. The test data is used to determine how well the machine can predict based on its previous learning. The training and validation losses, as well as the training and validation accuracy, are calculated for each dataset. The error on the training set of data is known as training loss. Validation loss is the error that occurs when the trained network processes the validation set of data. Testing accuracy accuracy we receive when we apply the model to the testing data, while training accuracy is the accuracy we get when we apply the model to the training data.

We also have generated Confusion matrix for our model. It’s a table which need to explain the performance of a method on test dataset. Confusion matrix shown the assumed errors of this model. Classification report have also generated from confusion matrix. Here, it shows the models accuracy, precision, recall, F1score and support. These are described shortly here.

**Precision:**

The closeness of at least two measurements to one another is alluded by Precision. It is free from exactness.

Precision = TP/(TP + FP) …………………………..(4.1)

**Recall:**

Recall refers to a classifier's capacity to detect every positive case. For each class, the ratio of real positives to false - negative and true positive rate is computed.

Recall = TP/(TP+FN) ………………………….(4.2)

**F1 Score**

The F1 score is the weighted harmonic mean of accuracy, with 1.0 denoting the highest and 0.0 denoting the poorest.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)……...(4.3)

**Support**

Support is the amount of instances of the class that actually occur in the dataset that is provided; it does not vary between models but rather diagnoses the evaluation process.

**Accuracy**

The most basic model performance measure is accuracy, which is defined as the proportion of properly predicted to total observations.

Accuracy =(TP + TN) / (TP + FN + TN + FP ) …………………………….(4.4)

### 4.5.1 Performance Analysis of DenseNet-121 using our Skin cancer dataset

**TABLE 4.1: Training Accuracy and Validation Accuracy With number Of Epochs**

| Number Of Epoch | Training Accuracy | Validation Accuracy |
|---|---|---|
| 1 | 0.79 | 0.65 |
| 5 | 0.86 | 0.84 |
| 10 | 0.8927 | 0.89 |
| 15 | 0.92 | 0.90 |

Figure 4.1 displays the model's training and validation accuracy for the face dataset. We can observe that training and validation accuracy are increasing as the number of epoch increases. Our training data's cost function value is loss, whereas the cross-validation data's cost function value is val loss. Both losses (loss and val loss) are falling, while both acc (acc and val acc) are increasing. As a result, this implies that the modeling has been properly trained. The val acc is a metric for how accurate our model's predictions are.

```
 accuracy: 0.9003 - precision: 0.8866 - recall: 0.8947 - f1: 0.6617 -

cy: 0.9181 - precision: 0.9139 - recall: 0.9048 - f1: 0.6606 - mc: 0.6
05.
 accuracy: 0.9181 - precision: 0.9139 - recall: 0.9048 - f1: 0.6606 -

cy: 0.9204 - precision: 0.9130 - recall: 0.9114 - f1: 0.6593 - mc: 0.6
-05.
 accuracy: 0.9204 - precision: 0.9130 - recall: 0.9114 - f1: 0.6593 -

cy: 0.9283 - precision: 0.9207 - recall: 0.9215 - f1: 0.6588 - mc: 0.6
e-05.
 accuracy: 0.9283 - precision: 0.9207 - recall: 0.9215 - f1: 0.6588 -

cy: 0.9249 - precision: 0.9118 - recall: 0.9240 - f1: 0.6579 - mc: 0.6
-06.
 accuracy: 0.9249 - precision: 0.9118 - recall: 0.9240 - f1: 0.6579 -
```

```
val_accuracy: 0.9061 - val_precision: 0.9161 - val_recall: 0.8733 - val_f1: 0.6543 ·

val_accuracy: 0.8985 - val_precision: 0.8923 - val_recall: 0.8833 - val_f1: 0.6586 ·

val_accuracy: 0.9015 - val_precision: 0.8904 - val_recall: 0.8933 - val_f1: 0.6565 ·

al_accuracy: 0.8909 - val_precision: 0.8519 - val_recall: 0.9200 - val_f1: 0.6550 - \

/al_accuracy: 0.8985 - val_precision: 0.8746 - val_recall: 0.9067 - val_f1: 0.6543 -
```

Figure 4.1 : The training accuracy and the validation accuracy

### 4.5.2 The confusion matrix

The confusion matrix of the evaluated results is presented in Fig 4.2.

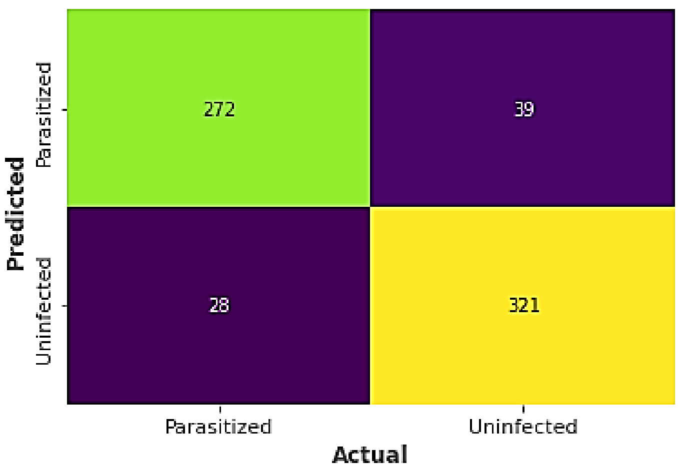


Figure 4.2: Confusion matrix

## 4.6 DISCUSSION

The Proposed model gives better results for our datasets. It does not misclassify any skin cancer. The final accuracy of our approached model is 92% to the skin disease dataset. At first, our accuracy comes to 80%. Then again we update the data set. Then accuracy comes to 86%. Some more work with Densenet where accuracy came 64%, 77.9%, 77.3%, 80%. But we have increased our data set and our accuracy has improved.

# CHAPTER V

# CONCLUSION AND FUTURE WORKS

## 5.1 CONCLUSION

In our work, the DenseaNet-121 CNN architecture has been proposed. The proposed DenseaNet-121 automatically recognizes and identifies a variety of skin cancers with speed, accuracy, and consistency. We also used various types of skin disease datasets to compare our proposed model to existing work, such as Resnet, U-net, VGGnet, and ImageNet.

Our suggested model performs well on the skin disease dataset, with an accuracy of 92%. We discussed the performance comparison in our thesis work. So, it is well understood that the DenseaNet-121 model which we proposed works better than the other model.

## 5.2 CONTRIBUTION OF THIS THESIS

No one has ever worked with DenseaNet-121 before. First we work with 2 classes datasets which are benign and malignant where we get good accuracy. Next we expand the dataset and we check in many images how our model is doing good. I checked and found that we got good accuracy even after adding new dataset.

## 5.3 FUTURE WORKS

Some aspects of the present research work can be further investigated and improved. The following recommendations are made based on the literature evaluations and studies undertaken in this thesis:

In the future, work will be done to improve DenseaNet-121 s accuracy rate by using more datasets and to improve the system's robustness. If we can do this, we will be able to research more in the future. If we can detect skin diseases image properly and accuracy is good, then we will try to apply more algorithm later.

# REFERENCES


1. Javaid, M. Sadiq and F. Akram, "Skin Cancer Classification Using Image Processing and Machine Learning," 2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST), 2021, pp. 439-444, doi:

10.1109/IBCAST51254.2021.9393198.

2. R. Desai, J. Panchal, A. Mitra, S. Mamlekar, H. Chari and S. Aswale, "Skin Cancer Detection Using Machine Learning: A Survey," 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM), 2022, pp. 31-35, doi:

10.1109/ICIEM54221.2022.9853084.

3. M. Ramachandro, T. Daniya and B. Saritha, "Skin Cancer Detection Using Machine Learning Algorithms," 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), 2021, pp. 1-7, doi: 10.1109/i-PACT52855.2021.9696874.

4. S. Labde and N. Vanjari, "Prediction of Skin Cancer Using CNN," 2022 3rd International Conference for Emerging Technology (INCET), 2022, pp. 1-4, doi: 10.1109/INCET54531.2022.9825093.

5. M. Vidya and M. V. Karki, "Skin Cancer Detection using Machine Learning Techniques," 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 2020, pp. 1-5, doi:

10.1109/CONECCT50063.2020.9198489.

6. P. Singh, M. Kumar and A. Bhatia, "A Comparative Analysis of Deep Learning Algorithms for Skin Cancer Detection," 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), 2022, pp. 1160-1166, doi:

10.1109/ICICCS53718.2022.9788197.

7. A. -R. H. Ali, J. Li and G. Yang, "Automating the ABCD Rule for Melanoma Detection: A Survey," in IEEE Access, vol. 8, pp. 83333-83346, 2020, doi: 10.1109/ACCESS.2020.2991034.

8. T. Guergueb and M. A. Akhloufi, "Multi-Scale Deep Ensemble Learning for Melanoma Skin Cancer Detection," 2022 IEEE 23rd International Conference on Information Reuse and Integration for Data Science (IRI), 2022, pp. 256-261, doi:

10.1109/IRI54793.2022.00063.

9. L. Gaur, U. Bhatia and S. Bakshi, "Cloud Driven Framework for Skin Cancer Detection using Deep CNN," 2nd

International Conference on Innovative Practices in Technology and Management (ICIPTM), 2022, pp. 460-464, doi: 2022

10.1109/ICIPTM54933.2022.9754216.

10. L. D. Biasi, A. A. Citarella, M. Risi and G. Tortora, "A Cloud Approach for Melanoma Detection Based on Deep Learning Networks," in IEEE Journal of Biomedical and Health Informatics, vol. 26, no. 3, pp. 962-972, March 2022, doi:

10.1109/JBHI.2021.3113609.

11. “Global Cancer Facts & Figures,” American Cancer Society. [Online]. Available:

https://www.cancer.org/research/cancer-facts-statistics/global.html

12. “Skin Cancer Facts & Statistics,” Skin Cancer Foundation. [Online].

Available: https://www.skincancer.org/skin-cancer-information/skin-cancerfacts

13. “Common Cancer Types,” National Cancer Institute. [Online]. Available:

https://www.cancer.gov/types/common-cancersE

14. R. L. Siegel, K. D. Miller and A. Jemal, “Cancer Statistics, 2019,” CA:A Cancer

Journal for Clinicians, vol. 69, pp. 7-34, 2019.

15. H. Kittler, H. Pehamberger, K. Wolff and M. Binder, “Diagnostic accuracy of

dermoscopy,” The Lancelet Oncology, vol. 3, no. 3, pp. 159-65, 2002.

16. “Man against machine: AI is better than dermatologists at diagnosing

skin cancer,” European Society for Medical Oncology. [Online]. Available:

www.sciencedaily.com/releases/2018/05/180528190839.html

17. S. Amutha and R. M. Devi, “Early detection of malignant melanoma using random forest algorithm,” International Journal for Trends in Engineering Technology, vol. 13, issue. 1, 2016

# Defense_report-final (31&36)

*by* Iiuc Central Library

# Defense_report-final (31&36)

ORIGINALITY REPORT

| 20% | 14% | 4% | 14% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to B.V. B College of Engineering and Technology, Hubli<br>Student Paper | 3% |
| 2 | epublications.regis.edu<br>Internet Source | 2% |
| 3 | Submitted to Benowa State High School<br>Student Paper | 1% |
| 4 | Submitted to University of Wales Institute, Cardiff<br>Student Paper | 1% |
| 5 | Submitted to Sheffield Hallam University<br>Student Paper | 1% |
| 6 | Submitted to Queen's University of Belfast<br>Student Paper | 1% |
| 7 | utpedia.utp.edu.my<br>Internet Source | 1% |
| 8 | Submitted to Liverpool John Moores University<br>Student Paper | <1% |

| | | |
|---|---|---|
| 9 | Submitted to Jordan University of Science & Technology<br>Student Paper | <1 % |
| 10 | Submitted to Yeditepe University<br>Student Paper | <1 % |
| 11 | Submitted to Universiti Tenaga Nasional<br>Student Paper | <1 % |
| 12 | www.researchgate.net<br>Internet Source | <1 % |
| 13 | Submitted to PSG Institute of Management Coimbatore<br>Student Paper | <1 % |
| 14 | pdfcoffee.com<br>Internet Source | <1 % |
| 15 | library.samdu.uz<br>Internet Source | <1 % |
| 16 | Submitted to Nottingham Trent University<br>Student Paper | <1 % |
| 17 | Submitted to University of North Texas<br>Student Paper | <1 % |
| 18 | www.slideshare.net<br>Internet Source | <1 % |
| 19 | Arise Roja Ganga Bhavani, Kantheti Leela, Medasani Pravalika, Kotha Sita Kumari. "Speech emotion recognition using both deep | <1 % |

learning and machine learning techniques", AIP Publishing, 2022
Publication

| | | |
|---|---|---|
| 20 | Submitted to Asia Pacific University College of Technology and Innovation (UCTI)<br>Student Paper | <1 % |
| 21 | Submitted to Midlands State University<br>Student Paper | <1 % |
| 22 | Submitted to University of Louisiana, Lafayette<br>Student Paper | <1 % |
| 23 | sutir.sut.ac.th:8080<br>Internet Source | <1 % |
| 24 | Submitted to Higher Education Commission Pakistan<br>Student Paper | <1 % |
| 25 | Submitted to Nepal College of Information Technology<br>Student Paper | <1 % |
| 26 | link.springer.com<br>Internet Source | <1 % |
| 27 | Submitted to Universiti Kebangsaan Malaysia<br>Student Paper | <1 % |
| 28 | Submitted to Universiti Teknikal Malaysia Melaka<br>Student Paper | <1 % |

| | | | |
|---|---|---|---|
| 29 | ijritcc.org<br>Internet Source | <1% |
| 30 | Submitted to Bournemouth University<br>Student Paper | <1% |
| 31 | Submitted to Lincoln Land Community College<br>Student Paper | <1% |
| 32 | apothesis.lib.hmu.gr<br>Internet Source | <1% |
| 33 | Iqbal H. Sarker. "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions", SN Computer Science, 2021<br>Publication | <1% |
| 34 | Chenchireddygari Sudharshan Reddy, Adoni Salauddin Siddiq, N. Jayapandian. "Machine Learning based Loan Eligibility Prediction using Random Forest Model", 2022 7th International Conference on Communication and Electronics Systems (ICCES), 2022<br>Publication | <1% |
| 35 | Submitted to Visvesvaraya Technological University, Belagavi<br>Student Paper | <1% |
| 36 | iq.opengenus.org<br>Internet Source | <1% |

| | | |
|---|---|---|
| 37 | Ashutosh Lembhe, Pranav Motarwar, Rudra Patil, Susan Elias. "Enhancement in Skin Cancer Detection using Image Super Resolution and Convolutional Neural Network", Procedia Computer Science, 2023<br>Publication | <1 % |
| 38 | singularityhub.com<br>Internet Source | <1 % |
| 39 | www.arxiv-vanity.com<br>Internet Source | <1 % |
| 40 | Submitted to International Islamic University Chittagong<br>Student Paper | <1 % |
| 41 | cancerandme.info<br>Internet Source | <1 % |
| 42 | www.thefreelibrary.com<br>Internet Source | <1 % |
| 43 | Submitted to Universiti Malaysia Perlis<br>Student Paper | <1 % |
| 44 | Submitted to University of Northumbria at Newcastle<br>Student Paper | <1 % |
| 45 | bmcbioinformatics.biomedcentral.com<br>Internet Source | <1 % |
| 46 | scholarworks.iupui.edu<br>Internet Source | |

<1 %

| | | |
|---|---|---|
| 47 | www.igi-global.com<br>Internet Source | <1 % |
| 48 | Submitted to Cambridge International University<br>Student Paper | <1 % |
| 49 | biblio.ugent.be<br>Internet Source | <1 % |
| 50 | pastel.archives-ouvertes.fr<br>Internet Source | <1 % |
| 51 | "Machine Learning Algorithms based Skin Disease Detection", International Journal of Innovative Technology and Exploring Engineering, 2019<br>Publication | <1 % |
| 52 | dione.lib.unipi.gr<br>Internet Source | <1 % |
| 53 | dspace.bracu.ac.bd:8080<br>Internet Source | <1 % |
| 54 | eprints.utm.my<br>Internet Source | <1 % |
| 55 | macsphere.mcmaster.ca<br>Internet Source | <1 % |

neuroquantology.com

| | | |
|---|---|---|
| 56 | Internet Source | <1% |
| 57 | nrl.northumbria.ac.uk<br>Internet Source | <1% |
| 58 | ourspace.uregina.ca<br>Internet Source | <1% |
| 59 | acikbilim.yok.gov.tr<br>Internet Source | <1% |
| 60 | dspace.daffodilvarsity.edu.bd:8080<br>Internet Source | <1% |
| 61 | eprints.utar.edu.my<br>Internet Source | <1% |
| 62 | hiof.brage.unit.no<br>Internet Source | <1% |
| 63 | 100-actuaires.institutdesactuaires.com<br>Internet Source | <1% |
| 64 | 1library.net<br>Internet Source | <1% |
| 65 | Submitted to ABV-Indian Institute of Information Technology and Management Gwalior<br>Student Paper | <1% |
| 66 | Submitted to British University in Egypt<br>Student Paper | <1% |

| | | |
|---|---|---|
| 67 | catalog.libraries.psu.edu<br>Internet Source | <1% |
| 68 | elibrary.buse.ac.zw:8080<br>Internet Source | <1% |
| 69 | essay.utwente.nl<br>Internet Source | <1% |
| 70 | mafiadoc.com<br>Internet Source | <1% |
| 71 | ruor.uottawa.ca<br>Internet Source | <1% |
| 72 | upcommons.upc.edu<br>Internet Source | <1% |
| 73 | www.scpe.org<br>Internet Source | <1% |
| 74 | towardsdatascience.com<br>Internet Source | <1% |

| | | | |
|---|---|---|---|
| Exclude quotes | On | Exclude matches | Off |
| Exclude bibliography | On | | |