

CATEGORIZATION OF INTEGUMENTARY SYSTEM DISORDERS USING DEEP LEARNING

Submitted in partial fulfillment of the requirements
for the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

M. Sai ram Asish (38110285)
Jakka Mithil Sai (38110210)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600 119**

MARCH – 2022

SATHYABAMA



INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with “A” grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **M. Sai ram asish (38110285)** and **J Mithil Sai (38110210)** who carried out the project entitled “**Categorization of integumentary system disorders using deep learning**” under my supervision from November 2021 to March 2022.

Internal Guide

Ms. R. Aishwarya M.E.

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **M. Sai ram asish (Reg No:38110285)** and **J Mithil Sai (Reg No: 38110210)** hereby declare that the Project Report entitled “**Categorization of integumentary system disorders using deep learning**” done by us under the guidance of **Ms. R. Aishwarya M.E.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in 2018-2022.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E.,Ph.D., Dean**, School of Computing **Dr.S.Vigneshwari M.E., Ph.D.** and **Dr.L.Lakshmanan M.E., Ph.D.** , Heads of the Department of Computer Science and Engineering for providing us necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and a deep sense of gratitude to my Project Guide **Ms. R. Aishwarya M.E.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

People in today's world are busy and occupied all of the time. We often overlook minor illnesses in our bodies as a result of our fast-paced lives. Skin disease is one of them. It is the most widespread disease on the planet. People generally consider dermatology problems as ephemeral, though that's not always the case. If the skin disease is not appropriately recognized, it might cause serious complications. The project's current models use segmentation techniques such as edge detection; however, our study seeks to diagnose skin disorders with high accuracy rates utilizing Deep learning. Our goal is to use CNN (convolutional neural networks) and transfer learning from the Inception model to identify and categorize skin disorders. This research seeks to inform victims of the potential repercussions and assist doctors in making an initial diagnosis. This is being implemented in both a web application and an Android application.

TABLE OF CONTENTS

Chapter No.	TITLE	Page No.
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	1
	1.1 OVERVIEW	1
	1.2 DEEP LEARNING	2
2	LITERATURE SURVEY	4
	2.1. RELATED WORK	4
3	AIM AND SCOPE OF THE PROJECT	6
	3.1. EXISTING SYSTEM	6
	3.2. PROPOSED SYSTEM	6
	3.3. OBJECTIVE	7
4	METHODOLOGY	8
	4.1. HARDWARE REQUIREMENTS	8
	4.2. SOFTWARE REQUIRMENTS	8
	4.3. SYSTEM ARCHITECTURE	8
	4.4. PYTHON AND JUPYTER	9
	4.5. DEEP LEARNING	16
	4.6. TRANSFER LEARNING	26
	4.7. ALGORITHMS AND ARCHITECURES	30
	4.7.1. CONVOLUTIONAL NEURAL NETWORK	30
	4.7.2. INCEPTION NEURAL NETWORK	36
	4.7.3. LIBRARIES AND FRAMEWORKS	43
	4.8. MODULES	51
	4.8.1. TRAINING AND TESTING THE MODEL	51
	4.8.2. DEPLOYING THE MODEL	51

5	RESULTS AND DISCUSSION	52
	5.1. PERFORMANCE ANALYSIS	52
6	CONCLUSION	53
	6.1. CONCLUSION	53
	REFERENCES	54
	APPENDICES	60
	A. SOURCE CODE	63
	B. SCREENSHOTS	117
	C. PLAGIARISM REPORT	120
	D. JOURNAL PAPER	121

LIST OF FIGURES

Fig No.	Figure Name	Page No.
4.3.1	SYSTEM ARCHITECTURE	9
4.5.1	ADVANTAGE OF DEEP LEARNING	17
4.5.3	SINGLE LAYER PERCEPTRON	18
4.5.3.2	WORKING OF A PERCEPTRON	20
4.5.4	MULTILAYER PERCEPTRON	21
4.5.4.2	FORWARD PROPAGATION	22
4.5.5.4.1	BACKWARD PROPAGATION	23
4.5.5.4.2	COMPUTING DERIVATIVES IN BACK PROPAGATION	24
4.6.1	PERFORMANCE OF TRANSFER LEARNING MODEL	27
4.6.2	ILLUSTRATION OF TRANSFER LEARNING	28
4.7.2.1.1	WORKING OF CONVOLUTIONAL LAYERS	32
4.7.2.1.2	PADDING IN CNN	33
4.7.2.1.3	ReLU GRAPH	34
4.7.2.1.4	MAX POOLING	35
4.7.3.1	INCEPTION ARCHITECTURE	36
4.7.3.2	A UNIT IN INCEPTION	40
4.7.3.3	1x1 CONVOLUTION	41
5.1	TRAINING ACCURACY OF THE MODEL	53

LIST OF ABBREVIATIONS

ABBREVIATIONS**EXPANSION**

ANN

Artificial Neural network

BGR

Blue Green Red

RGB

Decision Tree

CNN

Convolutional neural network

MLP

Multi Layer Perceptron

VGG

Very Deep Convolutional
Networks for Large-Scale Image
Recognition

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

AI is a broad term for intelligence exhibited by machines; we are witnessing a wide-reaching integration of AI into human life in the next few decades. Digital assistants, for example, help us with day-to-day tasks and give us information when we need it. Web searching is becoming increasingly accurate and efficient thanks to AI's predictive capabilities; an example is AI-powered web search. It has increased and can now give us accurate answers to our questions, which would have taken hours of research before. The disease detection or natural catastrophe forecasts are far more accurate than they were twenty years ago.

With the massive amount of data generated every day, it is natural that AI plays an essential role in the future of daily life. Artificial intelligence has been a topic of discussion for a long time, and it seems that it is finally reaching a point in which it can impact our day-to-day lives. Deep learning is a subdivision of AI. It aims at developing algorithms that analyze vast datasets to learn how to perform specific jobs.

Deep learning involves neural networks made up of multiple layers. These networks mimic the behaviour of the brain. Neural networks are at the core, and they provide the best results with unstructured data. This form of AI allows for more accessible training and better returns on investment. There are many advantages; one advantage is the high accuracy it provides with unstructured data. Deep learning outperforms machine learning when it comes to addressing complicated problems. The deep learning models' accuracy is high, and their predictions are close to humans. Most people do not know how to take care of it when it comes to skin. It is a sensitive and delicate organ that needs care and affection. An estimated 1 in 3 people have a skin disease at any time.

The human body has many potential entry points for parasites and bacteria to get inside and infect us with various ailments. Without a proper diagnosis, the disease will likely worsen. Some common skin diseases are eczema, psoriasis, acne, rosacea, and ichthyosis. Classifying integumentary system disorders using deep learning is one of the most challenging tasks for medical professionals. The accuracy of a computer-based diagnosis for a patient can help save time and money by providing a more accurate diagnosis. An important step in diagnosing a medical condition is to classify it as a certain type of disorder. This classification may be done using, for example, the ICD codes or the DSM-5 codes. To classify the group of International Classification of Diseases (ICD) codes into a suitable category, we use deep learning approach. Our results show that our method outperforms current methods in terms of both accuracy and efficiency.

Our goal is to four classify integumentary system disorders namely Acne, Psoriasis, Ringworm and Eczema using Deep learning. The model is trained using images of each disease as input. The images are fed to the deep learning layers, which adjust the weights of the neurons for every epoch. This model uses an Inception v3 neural network with 48 layers and an additional five dense layers for training photos of distinct classes. After completion of training the testing phase carried out to determine the accuracy of the model.

1.2 DEEP LEARNING

Deep learning allows computational models of multiple processing layers to learn and represent data with multiple levels of abstraction mimicking how the brain perceives and understands multimodal information, thus implicitly capturing intricate structures of large-scale data. Deep learning is a rich family of methods, encompassing neural networks, hierarchical probabilistic models, and a variety of unsupervised and supervised feature learning algorithms. The recent surge of interest in deep learning methods is due to the fact that they have been shown to outperform previous state-of-

the-art techniques in several tasks, as well as the abundance of complex data from different sources.

Convolutional neural networks (CNNs) or convnets, for short – have in recent years achieved results which were previously considered to be purely within the human realm. In this chapter we introduce CNNs, and for this we first consider regular neural networks, and how these methods are trained. After introducing the convolution, we introduce CNNs. They are very similar to the regular neural networks as they are also made up of neurons with learnable weights. But, in contrast to MLPs, CNNs make the explicit assumption that inputs have specific structure like images. This allows encoding this property into the architecture by sharing the weights for each location in the image and having neurons respond only locally.

CHAPTER 2

LITERATURE SURVEY

2.1 RELATED WORK

[2] Digital dermatology: Skin disease detection model using image processing, this paper uses algorithms like DCT (Direct Cosine Transform), SVD (Single Value Decomposition) and DWT (Direct Wavelet Transform) According to this paper, the attained accuracy for detecting skin disorders is 80%. The model's reliability would have increased greatly if neural networks had been utilized to train the images.

[7] Diagnosis of skin diseases using Convolutional Neural Networks, this paper solely relies on convolutional neural network (CNN) and achieved accuracy of seventy percent. The paper claims that accuracy can be improved if larger dataset is used. The model's accuracy would have improved significantly if the transfer learning approach had been used. Our model was also trained with around one thousand two hundred pictures of skin diseases and tested with two hundred images but achieved around ninety percent accuracy.

[3] Face Skin Disease Detection with Textural Feature Extraction, this paper aims at classifying skin face skin disease detection using K-nearest neighbour algorithm which produced an accuracy of eighty percent. If Deep learning is utilized and transfer learning is implemented then the efficiency of the model would have enhanced greatly.

Integumentary system disorders affect 71 percent of the world's population, almost three-fourths of humankind. Ailments of the integumentary system are the third most prevalent illness, and many who suffer from them do not consider them a significant problem that requires medical attention [2].

Depending on the skin state, the appearance of skin conditions in dermoscopy images might differ significantly. Hair, texture, and bubbles may further obfuscate the

distinction between skin infections and neighbouring clear skin. As a result, dermoscopy image classification is a complicated process [4]. We are deploying cutting-edge deep learning technologies to take on this complex task. When compared towards other ml algorithms, deep learning has been used to perform exceptionally challenging classification and segmentation jobs [32], [33] with great accuracy. The framework of the system is based primarily on convolutional layers [31]. Characteristics in the initial layers of deep learning network topologies are typically broad, while high-level features are often particular. Yosinski et al. [34] demonstrated that using model parameters transmitted from different jobs to initialize networks enhances results compared to arbitrary parameters [24].

CHAPTER 3

AIM AND SCOPE OF THE PROJECT

3.1 EXISTING SYSTEM

From the above literature survey, it can be inferred that some previous models used segmentation techniques with low accuracy, deep learning with CNN with less reliability and some models tried to achieve high accuracy with small dataset and fell flat.

The majority of the previous models concentrated on identifying whether the melanoma is benignant or malignant (due to abundance of datasets available on melanoma) and did not cover on other diseases like psoriasis, ringworm, etc.

3.2 PROPOSED SYSTEM

Our proposal is to make an interactive web application for the patient to get an instant diagnosis of the disease and a pdf with do's and dont's and a general medication to reduce the severity of the disease. Our model uses a small dataset and implements deep learning technology along with CNN and transfer learning.

The use of transfer learning using Inception v3 in the project sets apart from other models. Our model achieved accuracy on par with models which the dataset of ten thousand images with around thousand images.

3.3 OBJECTIVE

The project aims to perform diagnosis of integumentary system disorders like Ringworm, Melanoma, Psoriasis and Measles using deep learning employing

Convolutional Neural Networks (CNN) and transfer learning. The interactive website guides the patient to upload the photo of the affected area and with the use of our highly reliable deep learning model we display the disease name and a document with suggested medicines. The system in no way decided to replace doctors but it is to provide a preliminary diagnosis to the patients.

CHAPTER 4

METHODOLOGY

4.1. HARDWARE REQUIREMENTS

- GPU from AMD or NVIDIA is preferred to train the deep learning model.
- RAM: 8GB
- Processor: Intel Core i5
- Hard disk/SSD: 50 GB

4.2. SOFTWARE REQUIREMENTS

- Python version ≥ 3
- Jupyter Notebook (Can even be cloud platforms like Kaggle or Colab)
- Anaconda Navigator

4.3. SYSTEM ARCHITECTURE

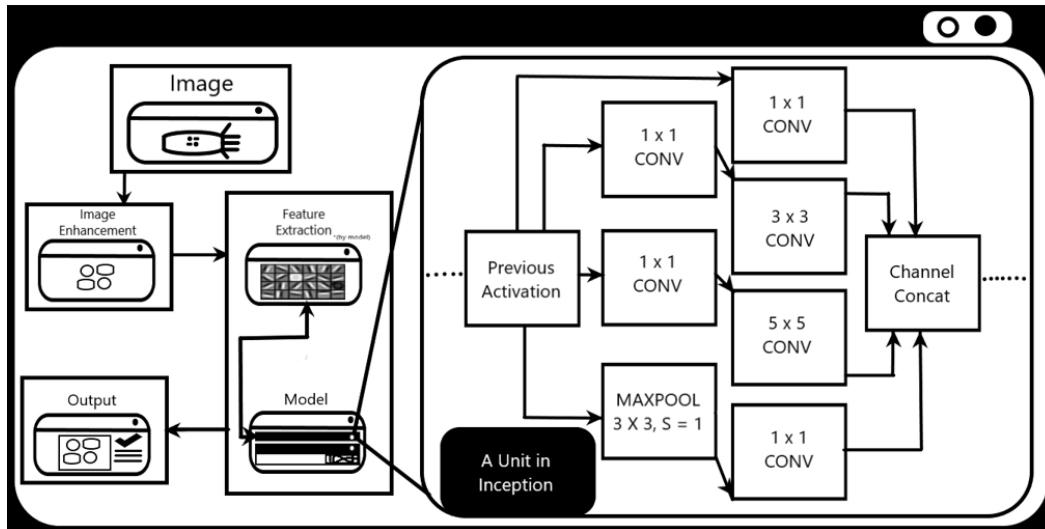


Fig. 4.3.1. System architecture

Steps:

1. The affected area of the captured picture is fed to the model.
2. The size of the image is adjusted to 224 x 224 resolution, brightness is slightly increased and the contrast between the colours is improved by using existing libraries in python.
3. The deep learning model is trained using CNN algorithm and transfer learning technique where training images for each disease are fed and the trained model is tested with the testing images for the diseases trained.
4. Desired accuracy of the model can be achieved by tweaking the hyper parameters such as learning rate, batch size, increasing the size of the dataset, adjusting the layers in neural network, removing or adding layers to the inception v3 network.
5. The trained model with desired accuracy is stored where the weights of all the layers are stored to instantly predict the class of a new picture.
6. The model then classifies the unknown image uploaded by the user.

4.4 PYTHON

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming.

It is often described as a "batteries included" language due to its comprehensive standard library. Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a garbage collection system using reference counting. Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible. Python 2 was discontinued with version 2.7.18 in 2020. Python consistently ranks as one of the most popular programming languages.

HISTORY

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to ABC programming language, which was inspired by SETL, capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989. Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's Benevolent Dictator for Life, a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. In January 2019, active Python core developers elected a 5-member "Steering Council" to lead the project.

As of 2021, the current members of this council are Barry Warsaw, Brett Cannon, Carol Willing, Thomas Wouters, and Pablo Galindo Salgado. Python 2.0 was released on 16 October 2000, with many major new features, including a cycle-detecting garbage collector and support for Unicode. Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely backward-compatible. Many of its major features were backported to Python 2.6.x and 2.7.x version series.

Releases of Python 3 include the 2 to 3 utility, which automates (at least partially) the translation of Python 2 code to Python 3. Python 2.7's end-of-life date was initially set at 2015 then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3. No more security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.6.x and later are supported. Python 3.9.2 and 3.8.8 were expedited as all versions of Python (including 2.7) had security issues, leading to possible remote code execution and web cache poisoning.

ANACONDA

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows us to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository. Anaconda. Now, if we are primarily doing data science work, Anaconda is also a great option. Anaconda is created by Continuum Analytics, and it is a Python distribution that comes preinstalled with lots of useful python libraries for data science.

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages and use multiple environments to separate these different versions. The command-line

program conda is both a package manager and an environment manager. This helps data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. We can use it to find the packages we want, install them in an environment, run the packages, and update them – all inside Navigator. Install Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution. Navigator allows us to launch common Python programs and easily manage conda packages, environments, and channels without using command-line commands.

Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. Anaconda comes with many built-in packages that we can easily find with `conda list` on our anaconda prompt. As it has lots of packages (many of which are rarely used), it requires lots of space and time as well. If we have enough space, time and do not want to burden ourself to install small utilities like JSON, YAML, it's better go for Anaconda. Conda : Conda is an open source, cross-platform, language-agnostic package manager and environment management system that installs, runs, and updates packages and their dependencies.

It was created for Python programs, but it can package and distribute software for any language (e.g., R), including multi-language projects. The conda package and environment manager is included in all versions of Anaconda, Miniconda, and Anaconda Repository. Anaconda is freely available, open source distribution of python and R programming languages which is used for scientific computations.

If we are doing any machine learning or deep learning project then this is the best place for us. It consists of many softwares which will help us to build our machine learning project and deep learning project. these softwares have great graphical user interface and these will make our work easy to do. We can also use it to run our python script. These are the software carried by anaconda navigator.

4.4.1 JUPYTER NOTEBOOK

This website acts as “meta” documentation for the Jupyter ecosystem. It has a collection of resources to navigate the tools and communities in this ecosystem, and to help us get started. Project Jupyter is a project and community whose goal is to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages". It was spun off from IPython in 2014 by Fernando Perez. Notebook documents are documents produced by the Jupyter Notebook App, which contain both computer code (e.g., python) and rich text elements (paragraph, equations, figures, links, etc....).

Notebook documents are both human-readable documents containing the analysis description and the results (figures, tables, etc.) as well as executable documents which can be run to perform data analysis. Installation: The easiest way to install the Jupyter Notebook App is installing a scientific python distribution which also includes scientific python packages.

The Jupyter Notebook App can be launched by clicking on the Jupyter Notebook icon installed by Anaconda in the start menu (Windows) or by typing in a terminal (cmd on Windows): “jupyter notebook” This will launch a new browser window (or a new tab) showing the Notebook Dashboard, a sort of control panel that allows (among other things) to select which notebook to open. When started, the Jupyter Notebook App can access only files within its start-up folder (including any sub-folder).

No configuration is necessary if we place our notebooks in our home folder or subfolders. Otherwise, we need to choose a Jupyter Notebook App start-up folder which will contain all the notebooks. Save notebooks: Modifications to the notebooks are automatically saved every few minutes. To avoid modifying the original notebook, make a copy of the notebook document (menu file -> make a copy...) and save the modifications on the copy. Executing a notebook: Download the notebook to execute the file and put it in the notebook folder (or a sub-folder of it).

- Launch the jupyter notebook app
- In the Notebook Dashboard navigate to find the notebook: clicking on its name will open it in a new browser tab.

- Click on the menu Help -> User Interface Tour for an overview of the Jupyter Notebook App user interface.
- We can run the notebook document step-by-step (one cell a time) by pressing shift + enter.
- We can run the whole notebook in a single step by clicking on the menu Cell -> Run All.
- To restart the kernel (i.e. the computational engine), click on the menu Kernel -> Restart. This can be useful to start over a computation from scratch (e.g. variables are deleted, open files are closed, etc...).

An IPYNB file is a notebook document created by Jupyter Notebook, an interactive computational environment that helps scientists manipulate and analyze data using Python.

The Jupyter Notebook App is a server-client application that allows editing and running notebook documents via a web browser. The Jupyter Notebook App can be executed on a local desktop requiring no internet access (as described in this document) or can be installed on a remote server and accessed through the internet.

In addition to displaying/editing/running notebook documents, the Jupyter Notebook App has a “Dashboard” (Notebook Dashboard), a “control panel” showing local files and allowing to open notebook documents or shutting down their kernels.

4.4.2 KERNEL

A notebook kernel is a “computational engine” that executes the code contained in a Notebook document. The ipython kernel, referenced in this guide, executes python code. Kernels for many other languages exist (official kernels). When we open a Notebook document, the associated kernel is automatically launched. When the notebook is executed (either cell-by-cell or with menu Cell -> Run All), the kernel performs the computation and produces the results. Depending on the

type of computations, the kernel may consume significant CPU and RAM. Note that the RAM is not released until the kernel is shut-down

4.4.3 NOTEBOOK DASHBOARD:

The Notebook Dashboard is the component which is shown first when we launch Jupyter Notebook App. The Notebook Dashboard is mainly used to open notebook documents, and to manage the running kernels (visualize and shutdown). The Notebook Dashboard has other features similar to a file manager, namely navigating folders and renaming/deleting files.

4.4.3.1 WORKING PROCESS:

- Download and install anaconda and get the most useful package for machine learning in Python.
- Load a dataset and understand its structure using statistical summaries and data visualization.
- Machine learning models, pick the best and build confidence that the accuracy is reliable.

Python is a popular and powerful interpreted language. Unlike R, Python is a complete language and platform that we can use for both research and development and developing production systems. There are also a lot of modules and libraries to choose from, providing multiple ways to do each task. It can feel overwhelming.

The best way to get started using Python for machine learning is to complete a project. It will force we to install and start the Python interpreter (at the very least). It will give us a bird's eye view of how to step through a small project. It will give us confidence, maybe to go on to run small projects. When we are applying machine learning to our own datasets, we are working on a project.

A machine learning project may not be linear, but it has a number of well-known steps:

- Define Problem.

- Prepare Data.
- Evaluate Algorithms.
- Improve Results.
- Present Results.

4.4.4 GOOGLE COLAB

This project's model is trained in google colab using the free GPU resource provided by Google for free of cost. Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup and the notebooks that we create can be simultaneously edited by the team members - just the way we edit documents in Google Docs. Colab supports many popular machine learning libraries which can be easily loaded in our notebook. These are the following advantages of google colabx`. Write and execute code in Python Document our code that supports mathematical equations Create/Upload/Share notebooks Import/Save notebooks from/to Google Drive Import/Publish notebooks from GitHub Import external datasets e.g. from Kaggle Integrate PyTorch, TensorFlow, Keras, OpenCV Free Cloud service with free GPU

4.5 DEEP LEARNING

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability—allowing it to “learn” from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy. Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services (such as digital assistants, voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars).

4.5.1 HISTORY

The ambition to create a system that simulates the human brain fueled the initial development of neural networks. In 1943, McCulloch and Pitt tried to understand how the brain could produce highly complex patterns by using interconnected basic cells, called neurons. The McCulloch and Pitts model of a neuron, called a MCP model, has made an important contribution to the development of artificial neural networks.

A series of major contributions in the field including LeNet and Long Short-Term Memory, leading up to today's "era of deep learning." One of the most substantial breakthroughs in deep learning came in 2006, when Hinton introduced the Deep Belief Network, with multiple layers of Restricted Boltzmann Machines, greedily training one layer at a time in an unsupervised way.

Guiding the training of intermediate levels of representation using unsupervised learning, performed locally at each level, was the main principle behind a series of developments that brought about the last decade's surge in deep architectures and deep learning algorithms.

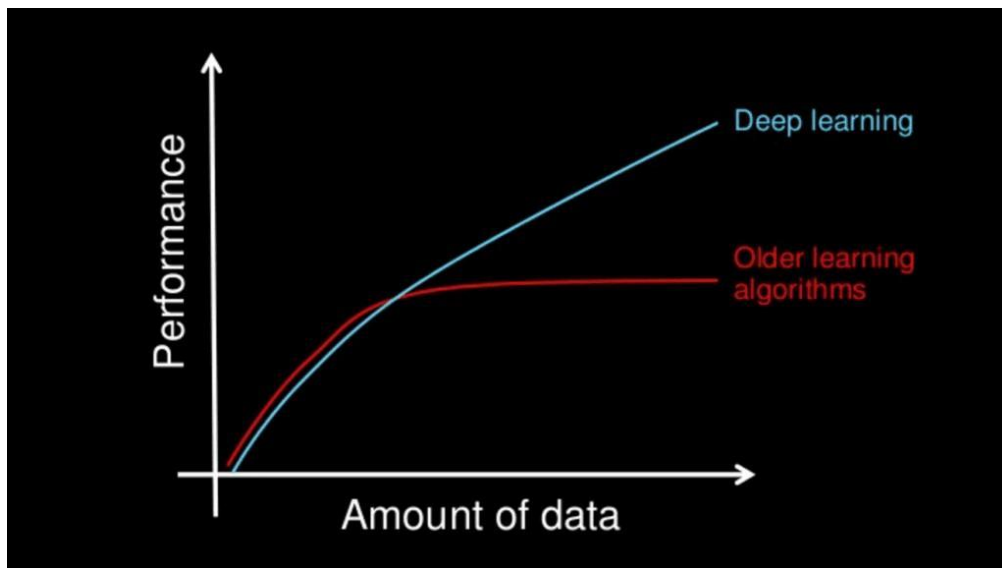


Fig.4.5.1. Advantage of deep learning

As show in the Fig.2., the machine learning algorithms like Decision tree, Random Forest, k-nearest neighbors algorithm, etc., as the amount of data increases the

performance of these algorithms become stagnant but deep learning algorithms like Convolutional Neural Networks (CNN), Long Term Short Memory (LSTM) networks accuracy improves as the data increases.

4.5.2 MACHINE LEARNING VS DEEP LEARNING

Machine learning is about computers being able to perform tasks without being explicitly programmed... but the computers still think and act like machines. Their ability to perform some complex tasks — gathering data from an image or video, for example — still falls far short of what humans are capable of.

Deep learning models introduce an extremely sophisticated approach to machine learning and are set to tackle these challenges because they've been specifically modeled after the human brain. Complex, multi-layered “deep neural networks” are built to allow data to be passed between nodes (like neurons) in highly connected ways. The result is a non-linear transformation of the data that is increasingly abstract. While it takes tremendous volumes of data to ‘feed and build’ such a system, it can begin to generate immediate results, and there is relatively little need for human intervention once the programs are in place.

Machine learning programs tend to be less complex than deep learning algorithms and can often run on conventional computers, but deep learning systems require far more powerful hardware and resources. This demand for power has driven increased use of graphical processing units. GPUs are useful for their high bandwidth memory and ability to hide latency (delays) in memory transfer due to thread parallelism (the ability of many operations to run efficiently at the same time).

Machine learning systems can be set up and operate quickly but may be limited in the power of their results. Deep learning systems take more time to set up but can generate results instantaneously (although the quality is likely to improve over time as more data becomes available).

4.5.3 PERCEPTRON

In artificial intelligence, the perceptron is an algorithm for supervised learning of binary classifiers. A binary classifier is a function which can decide whether or not

an input, represented by a vector of numbers, belongs to some specific class. It is a type of linear classifier, i.e., a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector.



Fig.4.5.3. Single layer perceptron

4.5.3.1 HISTORY

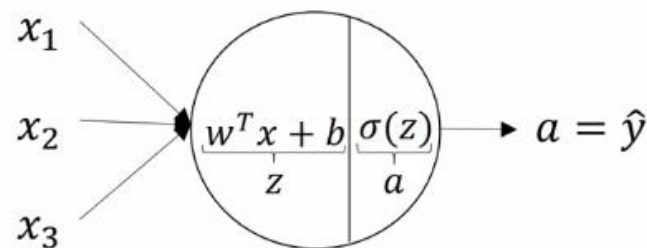
The perceptron algorithm was invented in 1958 at the Cornell Aeronautical Laboratory by Frank Rosenblatt, funded by the United States Office of Naval Research. The perceptron was intended to be a machine, rather than a program, and while its first implementation was in software for the IBM 704, it was subsequently implemented in custom-built hardware as the "Mark 1 perceptron". This machine was designed for image recognition: it had an array of 400 photocells, randomly connected to the "neurons". Weights were encoded in potentiometers, and weight updates during learning were performed by electric motors.

In a 1958 press conference organized by the US Navy, Rosenblatt made statements about the perceptron that caused a heated controversy among the fledgling AI community; based on Rosenblatt's statements, The New York Times reported the perceptron to be "the embryo of an electronic computer that [the

Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence”.

4.5.3.2 WORKING OF A PERCEPTRON

The inputs $x_1, x_2, x_3, x_4, \dots, x_n$ are multiplied with the weights of the layers $w_1, w_2, w_3, w_4, \dots, w_n$ to get $x_1 w_1, x_2 w_2, x_3 w_3, x_4 w_4, \dots, x_n w_n$. All these values are totaled i.e., $\sum_{i=1}^n w_i x_i + \sum_{i=1}^n b_i$ where b is the bias, this is denoted with the letter Z . Then it is further passed to the activation or threshold function which can be ReLU, Sigmoid, etc., which gives the final output as shown in Fig. 3., and Fig. 4.



$$z = w^T x + b$$

$$a = \sigma(z)$$

Fig. 4.5.4 Working of Perceptron

Single-layer perceptrons are only capable of learning linearly separable patterns. For a classification task with some step activation function, a single node will have a single line dividing the data points forming the patterns. More nodes can create more dividing lines, but those lines must somehow be combined to form more complex classifications. A second layer of perceptrons, or even linear nodes, are sufficient to solve a lot of otherwise non-separable problems.

4.5.4 MULTILAYER PERCEPTRON (MLP)

A multilayer perceptron is a neural network connecting multiple layers in a directed graph, which means that the signal path through the nodes only goes one way. Each node, apart from the input nodes, has a nonlinear activation function. An MLP uses backpropagation as a supervised learning technique. Since there are multiple layers of neurons, MLP is a deep learning technique. MLP is widely used for solving problems that require supervised learning as well as research into computational neuroscience and parallel distributed processing. Applications include speech recognition, image recognition and machine translation.

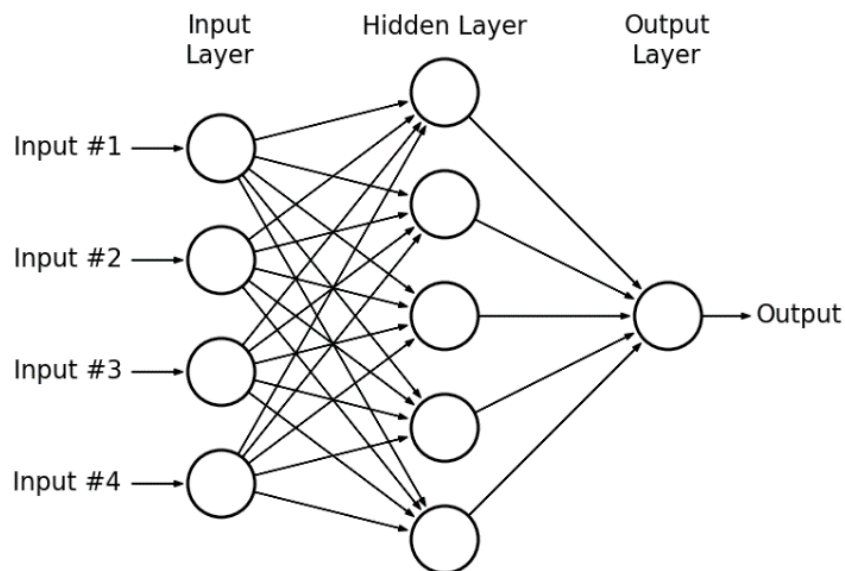


Fig. 4.5.4. Multilayer Perceptron

4.5.4.1 WORKING OF MULTILAYER PERCEPTRON

Just as with the perceptron, the inputs are pushed forward through the MLP by taking the dot product of the input with the weights that exist between the input layer and the hidden layer. This dot product yields a value at the hidden layer. We do not push this value forward as we would with a perceptron though. MLPs utilize activation functions at each of their calculated layers. There are many activation functions to discuss: rectified linear units (ReLU), sigmoid function, tanh. Push the calculated output at the

current layer through any of these activation functions. Once the calculated output at the hidden layer has been pushed through the activation function, push it to the next layer in the MLP by taking the dot product with the corresponding weights. Repeat steps two and three until the output layer is reached. At the output layer, the calculations will either be used for a backpropagation algorithm that corresponds to the activation function that was selected for the MLP (in the case of training) or a decision will be made based on the output (in the case of testing).

4.5.4.2 Forward Propagation:

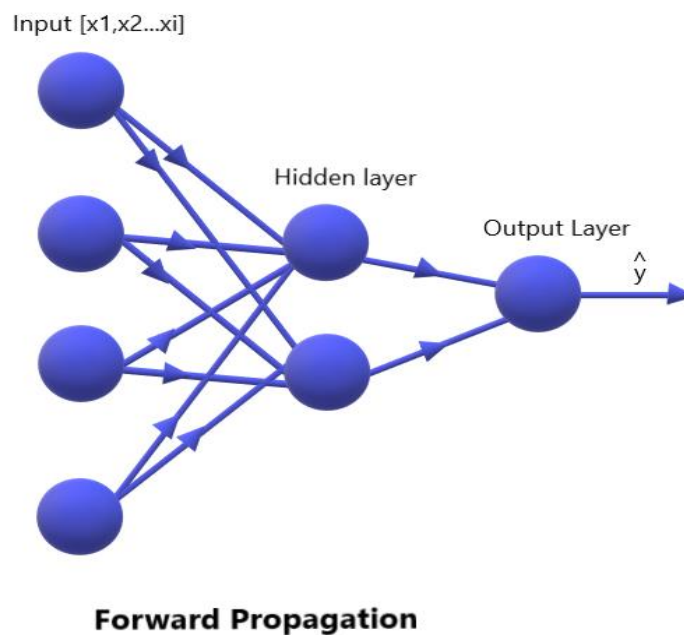


Fig. 4.5.4.2. Forward Propagation

Each hidden unit in every layer performs two operations. The first operation is calculating the dot product between previous layer activation (I_{l-1}) and weights $W(l)$ of current layer and this is summed up with the current layer bias $b(l)$, the result is stored in Z . In the second operation if it is for hidden layers ReLU activation is used and if it is for output layer then for single class classification sigmoid activation is used. To achieve

higher efficiency (for backpropagation), while performing first and second operation we store current parameters in cache.

The implementation of the forward propagation is divided into three functions. The first function is determined to apply logistic regression formula and return Z, cache. The second function is designed to calculate the activation function, it can be ReLU or sigmoid based on the stage at which it is called. This function returns A and cache. The third function is forward model this invokes the above two functions dynamically. This function returns AL and caches.

4.5.4.3 COMPUTING LOSS:

Calculating loss function gives us to the performance measure of the ANN. There are many types of loss functions, here we are using cross entropy loss function. Generally, this type of loss function is used for single class classification tasks.

Cross Entropy Loss (formula): $(1/\text{number of examples in the dataset}) * (-\text{dot_product}(\text{predicted_label}, \log(\text{previous_activation}).\text{transpose}) - \text{dot_product}(\text{1-predicted_label}, \log(1-\text{previous_activation}).\text{transpose}))$.

Cost function is a measure of how wrong the model is in terms of its ability to estimate the relationship between examples and the labels. This plays a crucial role in training a neural network. The whole objective of Artificial neural network is to minimize the cost. In other words, the lesser the cost of a model the more efficient it is. In the implementation the cost is calculated using Logistic loss and it is further squeezed i.e., the numbers with single dimensions are reduced to a simple number.

4.5.5.4 BACKPROPAGATION:

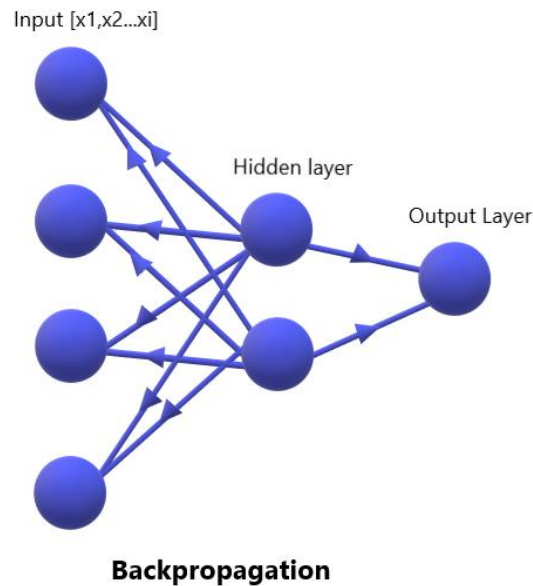


Fig. 4.5.5.4.1 BACKWARD PROPAGATION

Backprop algorithm became popular in the 1980s and now it is considered as backbone of neural network. Before this algorithm was introduced to predict a label a large structure i.e., huge layers and hidden units, Artificial neural network was used. After the usage of this algorithm the layers of ANN were reduced and the results were promising.

These derivatives can be later used for updating parameters in gradient descent. *Backpropagation* is the essence of neural net training. It is the practice of fine-tuning the weights of a neural net based on the error rate (i.e. loss) obtained in the previous epoch (i.e. iteration). Proper tuning of the weights ensures lower error rates, making the model reliable by increasing its generalization.

To apply backpropagation, we need to first compute derivatives of the da , dz , dw and db .

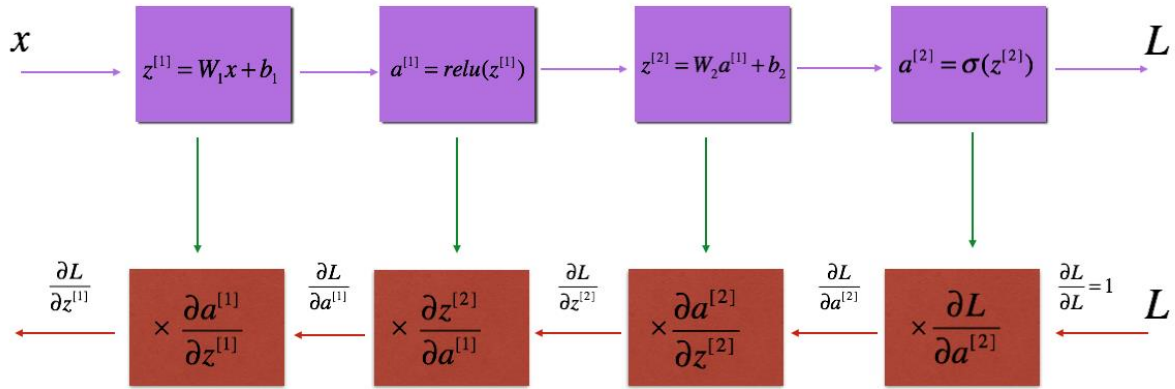


Fig. 4.5.5.4.2. Computing derivatives in Back propagation

For calculating da,

$$da = \frac{dL}{da}$$

$$da = \frac{-d}{da} (y \log(a) + (1 - y) \log \log(1 - a))$$

$$da = -\left(\frac{y}{a} + (1 - y) \left(\frac{1}{1 - a}\right)\right)$$

$$da = -\left(\frac{y}{a} - \frac{1 - y}{1 - a}\right)$$

$$da = \frac{-y}{a} + \frac{1 - y}{1 - a}$$

$$da = \frac{a - y}{a(1 - a)}$$

For calculating dz,

$$dz = \frac{dL}{dz}$$

Using chain rule,

$$dz = \frac{dL}{da} * \frac{da}{dz}$$

$$\frac{da}{dz} = \frac{d}{dz} \left(\frac{1}{1 + \exp \exp(-z)} \right)$$

$$\frac{da}{dz} = \frac{d}{dz} \left(\frac{e^z}{1 + e^z} \right)$$

Applying u/v rule,

$$\frac{da}{dz} = \frac{(1 + e^z)(e^z) - (e^z)(e^z)}{(1 + e^z)^2}$$

We know that $a = \frac{1}{1 + \exp(-z)}$,

$$\frac{da}{dz} = a - a^2$$

$$\frac{da}{dz} = a(1 - a)$$

$$dz = \frac{a - y}{a(1 - a)} * a(1 - a)$$

$$dz = a - y$$

For calculating dw,

$$\frac{dL}{dw} = \frac{dL}{dz} * \frac{dz}{dw}$$

$$\frac{dz}{dw} = \frac{d}{dw}(w \cdot x + b)$$

$$\frac{dz}{dw} = x$$

$$\frac{dL}{dw} = dz * x$$

For calculating db,

$$\frac{dL}{db} = \frac{dL}{dz} * \frac{dz}{db}$$

$$\frac{dz}{db} = \frac{d}{db}(w \cdot x + b)$$

$$\frac{dz}{db} = 1$$

$$\frac{dL}{db} = dz$$

Derivatives of da, dz, dw and db are $\frac{a-y}{a(1-a)}$, $a - y$, $dz * x$, dz respectively.

The Derivatives can be easily calculated using the cache we stored previously during forward propagation.

The implementation of backpropagation is split in to three different functions. The first function is linear backward it pulls the cache from forward propagation and applies the formulas for dw , db and da . The second function is linear activation backward, this function pulls cache and activation. The task of this function is to calculate the activation function ReLU or sigmoid dynamically. The third function is model backward, this also pulls the caches. Here all the gradients are calculated and these are returned in the form of dictionary.

4.6. TRANSFER LEARNING

Transfer learning, which focuses on transferring the knowledge across domains, is a promising machine learning methodology for solving the above problem. The concept about transfer learning may initially come from educational psychology. According to the generalization theory of transfer, as proposed by psychologist C.H. Judd, learning to transfer is the result of the generalization of experience. It is possible to realize the transfer from one situation to another, as long as a person generalizes his experience. According to this theory, the prerequisite of transfer is that there needs to be a connection between two learning activities.

4.6.1 EXAMPLES

In practice, a person who has learned the violin can learn the piano faster than others, since both the violin and the piano are musical instruments and may share some common knowledge. Inspired by human beings' capabilities to transfer knowledge across domains, transfer learning aims to leverage knowledge from a related domain (called source domain) to improve the learning performance or minimize the number of labeled examples required in a target domain.

It is worth mentioning that the transferred knowledge does not always bring a positive impact on new tasks. If there is little in common between domains, knowledge transfer could be unsuccessful. For example, learning to ride a bicycle cannot help us learn to play the piano faster. Besides, the similarities between domains do not always facilitate learning, because sometimes the similarities may be misleading. For example, although Spanish and French have a close

relationship with each other and both belong to the Romance group of languages, people who learn Spanish may experience difficulties in learning French, such as using the wrong vocabulary or conjugation. This occurs because previous successful experience in Spanish can interfere with learning the word formation, usage, pronunciation, conjugation, etc., in French.

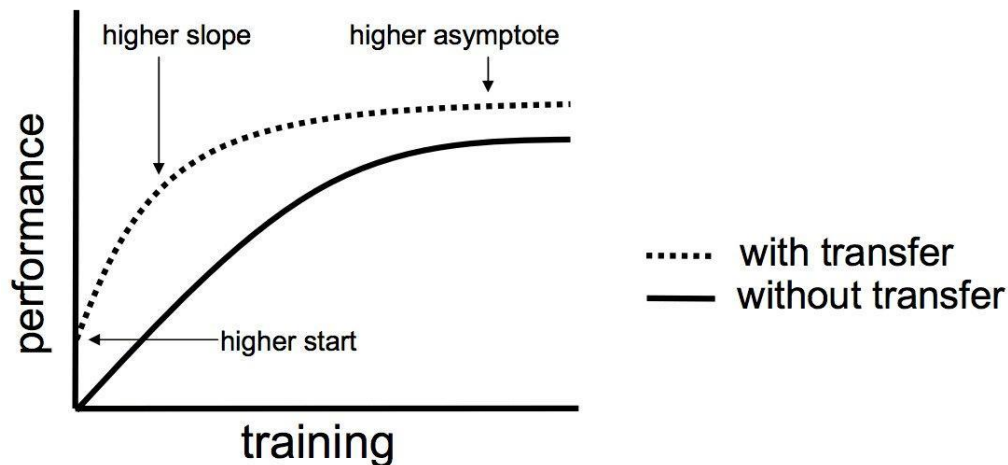


Fig. 4.6.1. PERFORMANCE OF TRANSFER LEARNING

4.6.2 HOW TRANSFER LEARNING WORKS

In transfer learning, the knowledge of an already trained machine learning model is applied to a different but related problem. For example, if we trained a simple classifier to predict whether an image contains a backpack, we could use the knowledge that the model gained during its training to recognize other objects like sunglasses. With transfer learning, we basically try to exploit what has been learned in one task to improve generalization in another. We transfer the weights that a network has learned at "task A" to a new "task B."

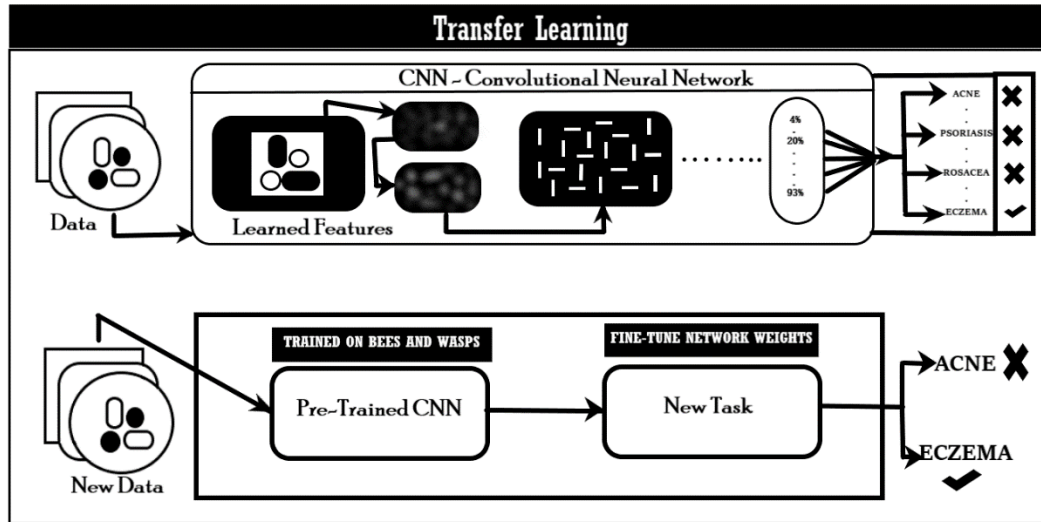


Fig. 4.6.2. ILLUSTRATION OF TRANSFER LEARNING

The general idea is to use the knowledge a model has learned from a task with a lot of available labeled training data in a new task that doesn't have much data. Instead of starting the learning process from scratch, we start with patterns learned from solving a related task. Transfer learning is mostly used in computer vision and natural language processing tasks like sentiment analysis due to the huge amount of computational power required. Transfer learning isn't really a machine learning technique, but can be seen as a "design methodology" within the field, for example, active learning. It is also not an exclusive part or study-area of machine learning. Nevertheless, it has become quite popular in combination with neural networks that require huge amounts of data and computational power.

Architecture	Top 1 accuracy	Top 5 accuracy
Alexnet	57%	80%
VGG	70%	91%
Inception v3	79%	95%

Table 1. Various CNN architectures

There are many popular neural networks architectures like Resnet, Inception, Xception, VGG, Alexnet, Lenet as shown in Table 1., but in this project we are making use of Inception v3 which has 24,000,000 trainable parameters and it is 47 layers deep. More than 1,000,000 images are trained and over 1000 classes are

categorized. In transfer learning we use a pretrained model which is fine tuned to get a desired rate of accuracy.

A pre-trained model is a saved network that was previously trained on a large dataset, typically on a large-scale image-classification task. We can either use the pretrained model as is or use transfer learning to customize this model to a given task. The intuition behind transfer learning for image classification is that if a model is trained on a large and general enough dataset, this model will effectively serve as a generic model of the visual world. We can then take advantage of these learned feature maps without having to start from scratch by training a large model on a large dataset.

There are two ways to customize pretrained model:

1.Feature Extraction:

Use the representations learned by a previous network to extract meaningful features from new samples. We simply add a new classifier, which will be trained from scratch, on top of the pretrained model so that we can repurpose the feature maps learned previously for the dataset. We do not need to retrain the entire model.

The base convolutional network already contains features that are generically useful for classifying pictures. However, the final, classification part of the pretrained model is specific to the original classification task, and subsequently specific to the set of classes on which the model was trained.

2. Fine-Tuning:

Unfreeze a few of the top layers of a frozen model base and jointly train both the newly-added classifier layers and the last layers of the base model. This allows us to "fine-tune" the higher-order feature representations in the base model in order to make them more relevant for the specific task. There is a general deep learning workflow.

1.Examine and understand the data

2. Build an input pipeline, in this case using Keras ImageDataGenerator (This is used in our project because it augments the number of images in our dataset since our dataset has only about one thousand two hundred trainable images)

3. Compose the model

- Load in the pretrained base model (and pretrained weights)
- Stack the classification layers on top

4. Train the model

5. Evaluate model

4.7 ALGORITHMS AND ARCHITECTURES

4.7.1. CONVOLUTIONAL NEURAL NETWORK (CNN)

Convolutional Neural Network has had ground breaking results over the past decade in a variety of fields related to pattern recognition; from image processing to voice recognition. The most beneficial aspect of CNNs is reducing the number of parameters in ANN. This achievement has prompted both researchers and developers to approach larger models in order to solve complex tasks, which was not possible with classic ANNs.

The most important assumption about problems that are solved by CNN should not have features which are spatially dependent. In other words, for example, in a face detection application, we do not need to pay attention to where the faces are located in the images. The only concern is to detect them regardless of their position in the given images. Another important aspect of CNN, is to obtain abstract features when input propagates toward the deeper layers. For example, in image classification, the edge might be detected in the first layers, and then the simpler shapes in the second layers, and then the higher-level features in the consequent layers.

4.7.2.1 STRUCTURE OF CNN

CNN has three main types of layers, which are:

- ☐ Convolutional layer
- ☐ Pooling layer
- ☐ Fully-connected (FC) layer

The convolutional layer is the first layer of a convolutional network. While convolutional layers can be followed by additional convolutional layers or pooling layers, the fully-connected layer is the final layer. With each layer, the CNN increases in its complexity, identifying greater portions of the image. Earlier layers focus on simple features, such as colors and edges. As the image data progresses through the layers of the CNN, it starts to recognize larger elements or shapes of the object until it finally identifies the intended object.

CONVOLUTIONAL LAYER

The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs. It requires a few components, which are input data, a filter, and a feature map. Let's assume that the input will be a color image, which is made up of a matrix of pixels in 3D. This means that the input will have three dimensions—a height, width, and depth—which correspond to RGB in an image. We also have a feature detector, also known as a kernel or a filter, which will move across the receptive fields of the image, checking if the feature is present. This process is known as a convolution.

The feature detector is a two-dimensional (2-D) array of weights, which represents part of the image. While they can vary in size, the filter size is typically a 3x3 matrix; this also determines the size of the receptive field. The filter is then applied to an area of the image, and a dot product is calculated between the input pixels and the filter. This dot product is then fed into an output array. Afterwards, the filter shifts by a stride, repeating the process until the kernel has swept across the entire image. The final output from the series of dot products from the input and the filter is known as a feature map, activation map, or a convolved feature.

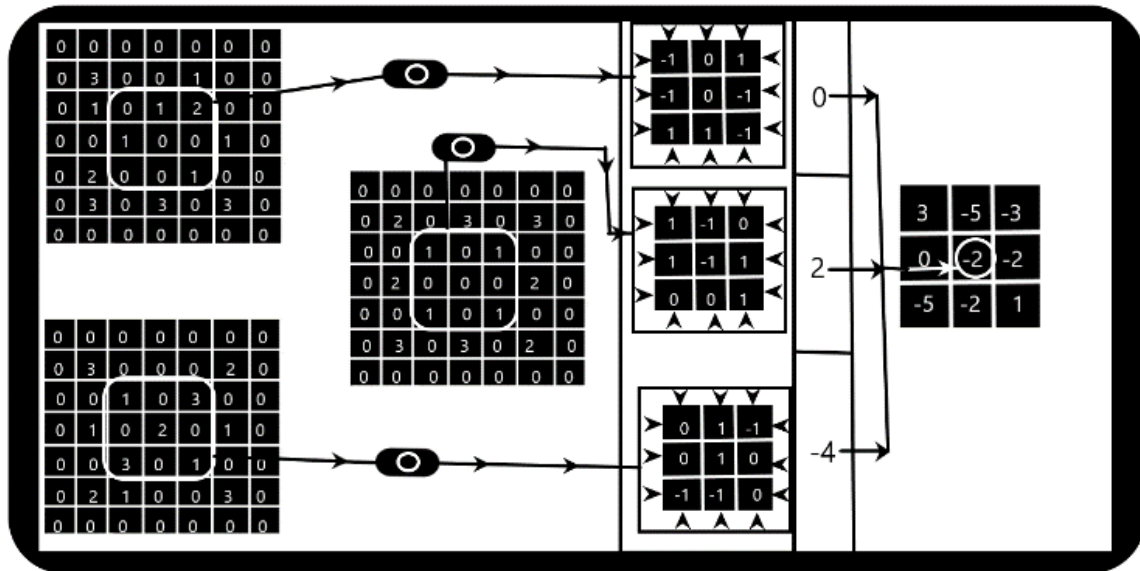


Fig. 4.7.2.1.1. Working of convolutional layers

Each output value in the feature map does not have to connect to each pixel value in the input image. It only needs to connect to the receptive field, where the filter is being applied. Since the output array does not need to map directly to each input value, convolutional (and pooling) layers are commonly referred to as “partially connected” layers. However, this characteristic can also be described as local connectivity. Note that the weights in the feature detector remain fixed as it moves across the image, which is also known as parameter sharing.

Some parameters, like the weight values, adjust during training through the process of backpropagation and gradient descent. However, there are three hyperparameters which affect the volume size of the output that need to be set before the training of the neural network begins. These include:

1. The number of filters affects the depth of the output. For example, three distinct filters would yield three different feature maps, creating a depth of three.

The general formula for calculating the output volume involves V is input volume size, F is kernel size, S is stride length, P being padding.

$$\frac{V - F + 2P}{S} + 1$$

2. Stride is the distance, or number of pixels, that the kernel moves over the input matrix. While stride values of two or greater is rare, a larger stride yields a smaller output.
3. Zero-padding is usually used when the filters do not fit the input image. This sets all elements that fall outside of the input matrix to zero, producing a larger or equally sized output. There are three types of padding:

Image

0	0	0	0	0	0	0
0						0
0						0
0						0
0						0
0						0
0	0	0	0	0	0	0

Fig. 4.7.2.1.2. Padding in CNN

- ☐ Valid padding: This is also known as no padding. In this case, the last convolution is dropped if dimensions do not align.
- ☐ Same padding: This padding ensures that the output layer has the same size as the input layer
- ☐ Full padding: This type of padding increases the size of the output by adding zeros to the border of the input.

After each convolution operation, a CNN applies a Rectified Linear Unit (ReLU) transformation to the feature map, introducing nonlinearity to the model.

ACTIVATION FUNCTION

Convolutional networks use an activation function, ReLU (Rectified Linear Activation Unit) function is generally used in most neural networks since it offers excellent performance and is easier to train the model. The general equation of ReLU is

$$Re(Z) = \max(0, Z)$$

The ReLU is a widely used activation function which doesn't change the value of the input if it is greater than zero and if the input value is less than zero it outputs the value as zero. The computation of cost for using this activation function is very low in comparison with other activation functions like sigmoid, tanh, etc. Plotting the graph for the below function where x is the input and $f(x)$ is the activation (ReLU) can be observed in Fig.8.

$$f(x) = \begin{cases} 0, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases}$$

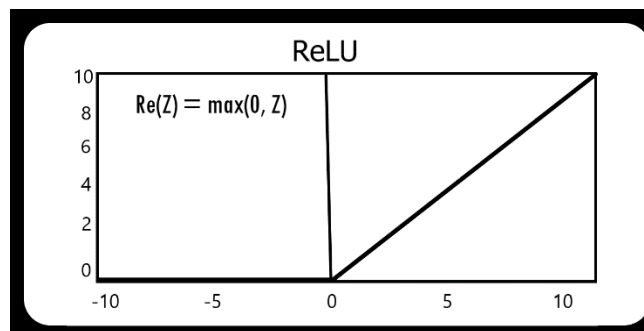


Fig. 4.7.2.1.3. ReLU graph

POOLING LAYER

Pooling layers aim to gradually reduce the dimensionality of the representation, and thus further reduce the number of parameters and the computational complexity of the model. The pooling layer operates over each activation map in the input, and scales its dimensionality using the “MAX” function. In most CNNs, these come in the form of max-pooling layers with kernels of a dimensionality of 2×2 applied with a stride of 2 along the spatial dimensions of the input.

This scales the activation map down to 25% of the original size - whilst maintaining the depth volume to its standard size. Due to the destructive nature of the pooling layer, there are only two generally observed methods of max-pooling. Usually, the stride and filters of the pooling layers are both set to 2×2 , which will allow the layer to extend through the entirety of the spatial dimensionality of the input. Furthermore, overlapping pooling may be utilised, where the stride is set to 2 with a kernel size set to 3.

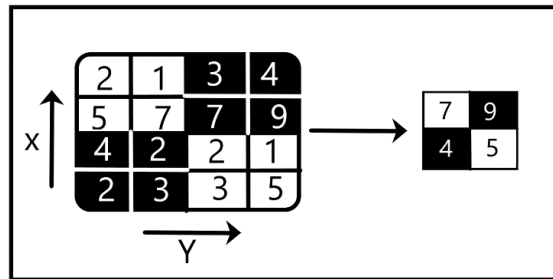


Fig. 4.7.2.1.4. Max pooling.

Due to the destructive nature of pooling, having a kernel size above 3 will usually greatly decrease the performance of the model. It is also important to understand that beyond max-pooling, CNN architectures may contain general-pooling. General pooling layers are comprised of pooling neurons that are able to perform a multitude of common operations including L1/L2-normalisation, and average pooling.

There are many types of pooling but these are the two popular types:

- Max pooling: As the filter moves across the input, it selects the pixel with the maximum value to send to the output array. As an aside, this approach tends to be used more often compared to average pooling.
- Average pooling: As the filter moves across the input, it calculates the average value within the receptive field to send to the output array.

As depicted in Fig.8, Max pooling is employed in the Inception network. Based on the stride length, the maximum value in the patch of input image is considered.

$$f_{(u,v)}(I) = m, n = 01MAX I_{(2U+m, 2V+n)}$$

In the above equation I is the matrix, U , V are the axes with 2×2 filter and number of pixels shifted also called as stride length is 2. The max of 2×2 filter with stride length of 2 is applied.

FULLY-CONNECTED LAYER

The name of the full-connected layer aptly describes itself. As mentioned earlier, the pixel values of the input image are not directly connected to the output layer in partially connected layers. However, in the fully-connected layer, each node in the output layer connects directly to a node in the previous layer. This layer performs the task of classification based on the features extracted through the previous layers and their different filters. While convolutional and pooling layers tend to use ReLU functions, FC layers usually leverage a softmax activation function to classify inputs appropriately, producing a probability from 0 to 1.

4.7.3. INCEPTION NEURAL NETWORK MODEL

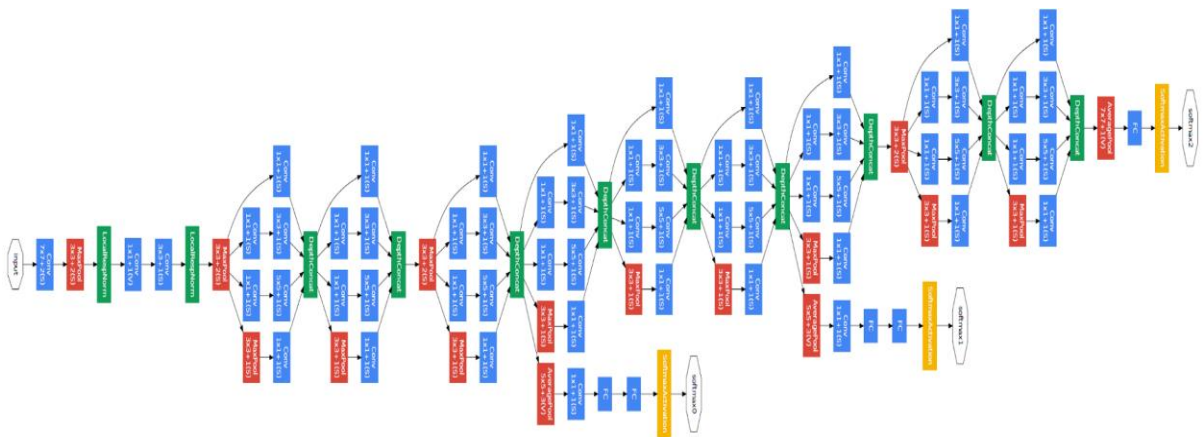


Fig. 4.7.3.1. Inception architecture

Inception Modules are used in Convolutional Neural Networks to allow for more efficient computation and deeper Networks through a dimensionality reduction with

stacked 1×1 convolutions. The modules were designed to solve the problem of computational expense, as well as overfitting, among other issues. The solution, in short, is to take multiple kernel filter sizes within the CNN, and rather than stacking them sequentially, ordering them to operate on the same level.

VGGNet VS INCEPTION (GoogLeNet)

Although VGGNet has the compelling feature of architectural simplicity, this comes at a high cost: evaluating the network requires a lot of computation. On the other hand, the Inception architecture of GoogLeNet was also designed to perform well even under strict constraints on memory and computational budget. For example, GoogleNet employed only 5 million parameters, which represented a $12\times$ reduction with respect to its predecessor AlexNet, which used 60 million parameters. Furthermore, VGGNet employed about $3\times$ more parameters than AlexNet.

The computational cost of Inception is also much lower than VGGNet or its higher performing successors. This has made it feasible to utilize Inception networks in big-data scenarios, where huge amount of data needed to be processed at reasonable cost or scenarios where memory or computational capacity is inherently limited, for example in mobile vision settings. It is certainly possible to mitigate parts of these issues by applying specialized solutions to target memory use or by optimizing the execution of certain operations via computational tricks. However, these methods add extra complexity. Furthermore, these methods could be applied to optimize the Inception architecture as well, widening the efficiency gap again.

VERSIONS OF INCEPTION V1, V2, V3

INCEPTION V1

The inception v1 was released in the year 2014 by Google which was initially called GoogLeNet. The problem statements for the Inception v1 architecture are as follows. Salient parts in the image can have extremely large variation in size. For instance, an image with a dog can be either of the following, as shown below. The area occupied by the dog is different in each image. Because of this huge variation

in the location of the information, choosing the right kernel size for the convolution operation becomes tough. A larger kernel is preferred for information that is distributed more globally, and a smaller kernel is preferred for information that is distributed more locally. Very deep networks are prone to overfitting. It is also hard to pass gradient updates through the entire network. Naively stacking large convolution operations is computationally expensive.

The solutions to the problems are as follows. Why not have filters with multiple sizes operate on the same level? The network essentially would get a bit “wider” rather than “deeper”. The authors designed the inception module to reflect the same. Naive inception module is used in the initial stage of Inception neural network architecture. It performs convolution on an input, with 3 different sizes of filters (1x1, 3x3, 5x5). Additionally, max pooling is also performed. The outputs are concatenated and sent to the next inception module.

Deep neural networks are computationally expensive. To make it cheaper, the authors limit the number of input channels by adding an extra 1x1 convolution before the 3x3 and 5x5 convolutions. Though adding an extra operation may seem counterintuitive, 1x1 convolutions are far cheaper than 5x5 convolutions, and the reduced number of input channels also help. The 1x1 convolution is introduced after the max pooling layer, rather than before.

GoogLeNet has 9 such inception modules stacked linearly. It is 22 layers deep (27, including the pooling layers). It uses global average pooling at the end of the last inception module. Needless to say, it is a pretty deep classifier. As with any very deep network, it is subject to the vanishing gradient problem. To prevent the middle part of the network from “dying out”, the authors introduced two auxiliary classifiers). They essentially applied softmax to the outputs of two of the inception modules, and computed an auxiliary loss over the same labels. The total loss function is a weighted sum of the auxiliary loss and the real loss. Weight value used in the paper was 0.3 for each auxiliary loss.

INCEPTION V2

Inception v2 architecture was presented in the very next year of Inception v1 i.e., 2015. The following are the hindrances to achieve higher accuracy. Reduce representational bottleneck. The intuition was that, neural networks perform better when convolutions didn't alter the dimensions of the input drastically. Reducing the dimensions too much may cause loss of information, known as a "representational bottleneck". Using smart factorization methods, convolutions can be made more efficient in terms of computational complexity.

The solutions to those problems are as follows. Factorize 5x5 convolution to two 3x3 convolution operations to improve computational speed. Although this may seem counterintuitive, a 5x5 convolution is 2.78 times more expensive than a 3x3 convolution. So stacking two 3x3 convolutions infact leads to a boost in performance..

Moreover, they factorize convolutions of filter size $n \times n$ to a combination of $1 \times n$ and $n \times 1$ convolutions. For example, a 3x3 convolution is equivalent to first performing a 1×3 convolution, and then performing a 3×1 convolution on its output. They found this method to be 33% more cheaper than the single 3x3 convolution. The filter banks in the module were expanded to remove the representational bottleneck. If the module was made deeper instead, there would be excessive reduction in dimensions, and hence loss of information.

INCEPTION V3

The Inception v3 was published in the same year of Inception v3. The v3 architecture is considered more reliable and standard than v2. In this project we are making use of Inception v3 which is over 48 layers deep, 24 million trainable parameters and trained over a million images. Problems resolved using Inception v3 architecture are as follows. The authors noted that the auxiliary classifiers didn't contribute much until near the end of the training process, when accuracies were nearing saturation. They argued that they function as regularizes, especially if they have Batch Normalization or Dropout operations. Possibilities to improve on the Inception v2 without drastically changing the modules were to be investigated.

The solutions to the above problems are as follows. RMSProp Optimizer. Factorized 7x7 convolutions. BatchNorm in the Auxillary Classifiers. Label Smoothing: A type of regularizing component added to the loss formula that prevents the network from becoming too confident about a class. Prevents over fitting.

THE WORKING OF INCEPTION NEURAL NETWORK

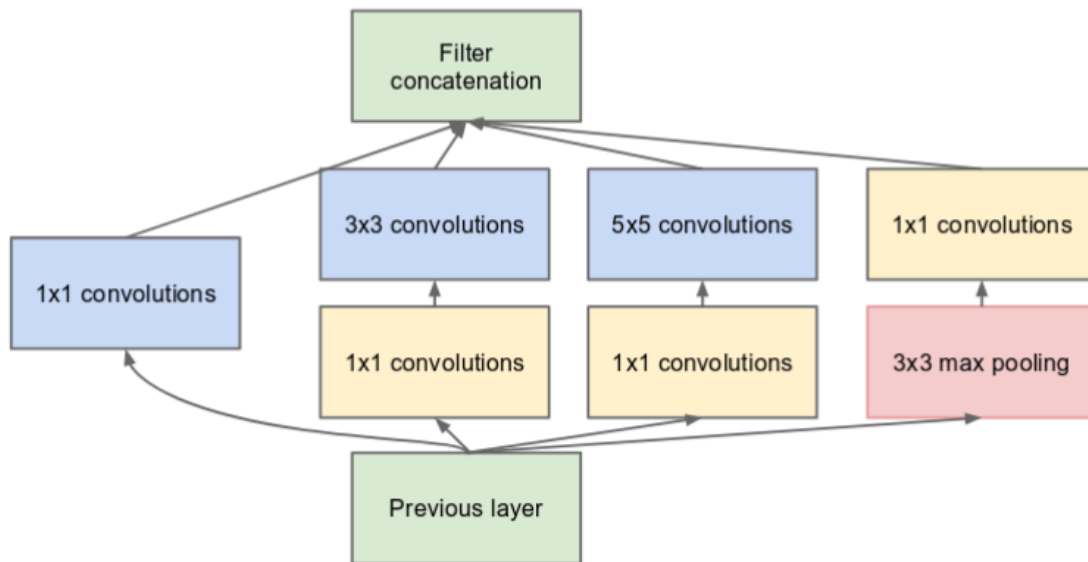


Fig. 4.7.3.2. A unit of Inception architecture

There are many similarities between the CNN like using Convolutional layer, performing pooling and Dense stage. The idea of Inception emerged what if multiple convolutions like 3x3, 1x1, 5x5 are combined with max pooling, Batch Normalization and each unit of this combination is connected to each other using Concat layers. This idea became very successful and gave the desired high accuracy to predict the images some time even overtaking human performance.

The following are the components of Inception neural network:

- ❖ A **convolutional operation** with 16 filters of size 1x1. The output tensor will be of size 32x32x16, where the last number refers to the number of resulting feature maps, equal to the number of filters convolved over the image.
- ❖ a **convolutional operation** with 32 filters of size 3x3. To maintain the output with the same width and height as the original feature maps, we can set padding=same and stride = 1. The output tensor will be of size 32x32x32.
- ❖ a **max-pooling operation** with a filter size of 3x3. The output tensor will be of size 32x32x64, in this case, since the pooling filter is passed over each feature map of the input tensor, the output tensor will have a depth equal to the original one = 64.
- ❖ **1x1 convolution** layers are used in the Inception network to decrease the complexity i.e., they are capable of reducing CNN output layers, for example, 8x8x22 can be reduced to 8x8x10, as shown in Fig. .

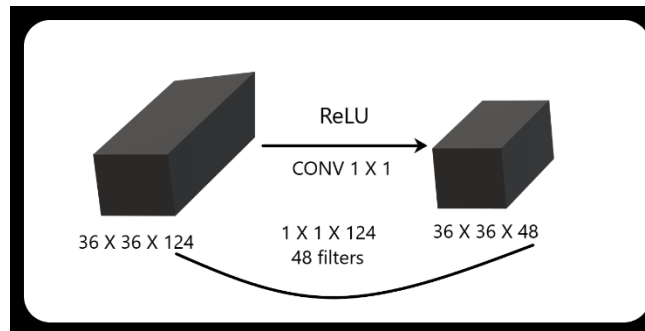


Fig. 4.7.3.3. 1x1 convolution

- ❖ **Batch Normalization** is the process of transforming the data to have a mean zero and standard deviation one. In this step we have our batch input from layer h, first, we need to calculate the mean of this hidden activation.

$$\mu = \frac{1}{m} \sum h_i$$

Here, m is the number of neurons at layer h.

The next step is to calculate the standard deviation of the hidden activations.

$$\sigma = \sqrt{\left[\frac{1}{m} \sum (h_i - \mu)^2 \right]}$$

We will normalize the hidden activations using these values. For this, we will subtract the mean from each input and divide the whole value with the sum of standard deviation and the smoothing term (ϵ).

The smoothing term(ϵ) assures numerical stability within the operation by stopping a division by a zero value. smoothing term(ϵ)

$$h_{i(norm)} = \frac{(h_i - \mu)}{\sigma + \epsilon}$$

Rescaling of Offsetting: In the final operation, the re-scaling and offsetting of the input take place. Here two components of the BN algorithm come into the picture, γ (gamma) and β (beta). These parameters are used for re-scaling (γ) and shifting(β) of the vector containing values from the previous operations.

$$h_i = \gamma h_{i(norm)} + \beta$$

These two are learnable parameters, during the training neural network ensures the optimal values of γ and β are used. That will enable the accurate normalization of each batch. The advantages of Batch normalization are it speeds up the training, optimizes the internal covariate shift and smoothens the loss function.

4.7.4 LIBRARIES AND FRAMEWORKS

The following are the libraries and frameworks used in this project

- Pillow module
- Matplotlib.pyplot module
- Os module
- Tensorflow v2.8.0 (Integrated with Keras)
- Numpy module

4.7.4.1 TENSORFLOW

TensorFlow is an open-source end-to-end platform for creating Machine Learning applications. It is a symbolic math library that uses dataflow and differentiable programming to perform various tasks focused on training and inference of deep neural networks. It allows developers to create machine learning applications using various tools, libraries, and community resources. Currently, the most famous deep learning library in the world is Google's TensorFlow. Google product uses machine learning in all of its products to improve the search engine, translation, image captioning or recommendations.

To give a concrete example, Google users can experience a faster and more refined search experience with AI. If the user types a keyword in the search bar, Google provides a recommendation about what could be the next word.

Google wants to use machine learning to take advantage of their massive datasets to give users the best experience. Three different groups use machine learning:

- Researchers
- Data Scientists
- Programmers

They can all use the same toolset to collaborate with each other and improve their efficiency.

Google does not just have any data; they have the world's most massive computer, so Tensor Flow was built to scale. TensorFlow is a library developed by the Google Brain Team to accelerate machine learning and deep neural network research.

It was built to run on multiple CPUs or GPUs and even mobile operating systems, and it has several wrappers in several languages like Python, C++ or Java.

HISTORY

A couple of years ago, deep learning started to outperform all other machine learning algorithms when giving a massive amount of data. Google saw it could use these deep neural networks to improve its services:

- Gmail
- Photo
- Google search engine

They build a framework called Tensorflow to let researchers and developers work together on an AI model. Once developed and scaled, it allows lots of people to use it.

It was first made public in late 2015, while the first stable version appeared in 2017. It is open source under Apache Open-Source license. We can use it, modify it and redistribute the modified version for a fee without paying anything to Google.

Next in this TensorFlow Deep learning tutorial, we will learn about TensorFlow architecture and how does TensorFlow work.

WORKING OF TENSORFLOW

TensorFlow enables us to build dataflow graphs and structures to define how data moves through a graph by taking inputs as a multi-dimensional array called Tensor. It allows us to construct a flowchart of operations that can be performed on these inputs, which goes at one end and comes at the other end as output.

TensorFlow Architecture

Tensorflow architecture works in three parts:

- Preprocessing the data
- Build the model
- Train and estimate the model

It is called Tensorflow because it takes input as a multi-dimensional array, also known as tensors. We can construct a sort of flowchart of operations (called a Graph) that we

want to perform on that input. The input goes in at one end, and then it flows through this system of multiple operations and comes out the other end as output.

This is why it is called TensorFlow because the tensor goes in it flows through a list of operations, and then it comes out the other side.

COMPATIBILITY OF TENSORFLOW

TensorFlow hardware, and software requirements can be classified into

Development Phase: This is when we train the model. Training is usually done on our Desktop or laptop.

Run Phase or Inference Phase: Once training is done TensorFlow can be run on many different platforms. We can run it on

- Desktop running Windows, macOS or Linux
- Cloud as a web service
- Mobile devices like iOS and Android

We can train it on multiple machines then we can run it on a different machine, once we have the trained model.

The model can be trained and used on GPUs as well as CPUs. GPUs were initially designed for video games. In late 2010, Stanford researchers found that GPU was also very good at matrix operations and algebra so that it makes them very fast for doing these kinds of calculations. Deep learning relies on a lot of matrix multiplication. TensorFlow is very fast at computing the matrix multiplication because it is written in C++. Although it is implemented in C++, TensorFlow can be accessed and controlled by other languages mainly, Python.

Finally, a significant feature of TensorFlow is the TensorBoard. The TensorBoard enables to monitor graphically and visually what TensorFlow is doing.

COMPONENTS OF TENSORFLOW

TENSOR

Tensorflow's name is directly derived from its core framework: Tensor. In Tensorflow, all the computations involve tensors. A tensor is a vector or matrix of n-dimensions that represents all types of data. All values in a tensor hold identical data type with a known (or partially known) shape. The shape of the data is the dimensionality of the matrix or array.

A tensor can be originated from the input data or the result of a computation. In TensorFlow, all the operations are conducted inside a graph. The graph is a set of computation that takes place successively. Each operation is called an op node and are connected to each other.

The graph outlines the ops and connections between the nodes. However, it does not display the values. The edge of the nodes is the tensor, i.e., a way to populate the operation with data.

GRAPHS

TensorFlow makes use of a graph framework. The graph gathers and describes all the series computations done during the training. The graph has lots of advantages:

- It was done to run on multiple CPUs or GPUs and even mobile operating system
- The portability of the graph allows to preserve the computations for immediate or later use. The graph can be saved to be executed in the future.
- All the computations in the graph are done by connecting tensors together
 - A tensor has a node and an edge. The node carries the mathematical operation and produces an endpoints output. The edges the edges explain the input/output relationships between nodes.

FEATURES OF TENSORFLOW

TensorFlow is the best library of all because it is built to be accessible for everyone. Tensorflow library incorporates different API to built at scale deep learning architecture

like CNN or RNN. TensorFlow is based on graph computation; it allows the developer to visualize the construction of the neural network with Tensorboard. This tool is helpful to debug the program. Finally, Tensorflow is built to be deployed at scale. It runs on CPU and GPU.

Tensorflow attracts the largest popularity on GitHub compare to the other deep learning framework.

The attributes of Tensorflow used in this project are (tf in the sense tensorflow):

- ❖ `tf.keras.preprocessing.image.ImageDataGenerator`
- ❖ `tensorflow.keras.applications.inception_v3`
- ❖ `tf.keras.optimizers.RMSprop`
- ❖ `tf.keras.layers.Dense`
- ❖ `tf.keras.layers.Flatten`
- ❖ `tf.keras.callbacks.ModelCheckpoint`
- ❖ `(Object from ImageDataGenerator).flow_from_directory`
- ❖ `tf.keras.callbacks.Callback`
- ❖ `tf.keras.callbacks.ModelCheckpoint`
- ❖ Learning rate=0.0001(RMSprop)
- ❖ `tf.keras.losses.CategoricalCrossentropy`

This framework in python is the backbone of this project.

Tensorflow is used in following aspects:

- Loading structures and weights of Inception v3 network.
- Developing layers in the model.
- Various hyper parameters like learning rate, number of epochs are adjusted.

- It supports multiple optimizers like RMSprop, Adam, SGD but in this project RMSprop is used.
- Activation functions like tanh, softmax, relu, linear, exponential are supported. For most of the layers relu is used and only for last layer of the model softmax is used since this is a multi-class classification.
- ImageDataGenerator function is used, where we can rotate, magnify, shift height and width of training images in an epoch. This is highly useful because it number of available training images for every epoch.
- We are making use of function `flow_from_directory` to directly sort the distinct classes with their training and testing datasets.
- Finally, we are compiling and fitting the model. The values of training accuracy and loss are stored in history variable which can be used to plot training accuracy. The testing accuracy can be calculated using `model.evaluate(validation_generator)`.

4.7.4.2 PILLOW

The Python Imaging Library (Pillow) adds image processing capabilities to our Python interpreter. This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities. The core image library is designed for fast access to data stored in a few basic pixel formats. It should provide a solid foundation for a general image processing tool.

OPENCV vs PILLOW

One of the huge setbacks of using opencv is that it reads the image in BGR (Blue Green Red), but the real-world images we observe are in RGB format. So, to we need to convert from BGR to RGB every time we pass the image to the model. Opencv is not beginner friendly in comparison with pillow. Pillow also supports multiple features like image contrast, image brightness, etc. Though Opencv is

faster than pillow, the scenario in this project is dependent on speed since we are using pillow only in the time of testing the image given by the user.

4.7.4.3 MATPLOTLIB.PYPILOT

Matplotlib is actually a cross platform software which is used to generate graphical diagrams but matplotlib.pyplot is the python version of this software which has all the features similar to the software. We can simply install the matplotlib through pip and import it as matplotlib.pyplot.

matplotlib.pyplot is a state-based interface to matplotlib. It provides an implicit, MATLAB-like, way of plotting. It also opens figures on our screen, and acts as the figure GUI manager. Pyplot is mainly intended for interactive plots and simple cases of programmatic plot generation. The various plots we can utilize using Pyplot are Line Plot, Histogram, Scatter, 3D Plot, Image, Contour, and Polar.

4.7.4.4 NUMPY

Numpy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors. NumPy is a NumFOCUS fiscally sponsored project.

We used in this project as `numpy.expand_dims(a, axis=0)` to expand the shape of an array. Insert a new axis that will appear at the axis position in the expanded array shape.

4.7.4.5 OS MODULE

This module provides a portable way of using operating system dependent functionality. If we just want to read or write a file see `open()`, if we want to manipulate paths, see the `os.path` module, and if we want to read all the lines in all the files on the command line see the `fileinput` module. In this project we used this module to change the directories in order to scan train, test and validate the images on the local machine.

4.7.4.6 FLASK

Flask is a web framework, it's a Python module that lets you develop web applications easily. It's has a small and easy-to-extend core: it's a microframework that doesn't include an ORM (Object Relational Manager) or such features. It does have many cool features like url routing, template engine. It is a WSGI web app framework.

WSGI

The Web Server Gateway Interface (Web Server Gateway Interface, WSGI) has been used as a standard for Python web application development. WSGI is the specification of a common interface between web servers and web applications.

WERKZEUG

Werkzeug is a WSGI toolkit that implements requests, response objects, and utility functions. This enables a web frame to be built on it. The Flask framework uses Werkzeug as one of its bases

We used Flask in the following aspects of our project

- We used Flask for model deployment.
- The static pages(html&css) are integrated using this module.
- Three user friendly web pages are deployed using flask: Introduction page to make the user understand about the application, Uploading page to upload the picture and reporting page to receive the report in pdf format.

4.8 MODULES

4.8.1 TRAINING AND TESTING THE MODEL

We trained the module with almost 1140 images and tested almost 140 images with the following is the split of training and testing set of the diseases namely, Ringworm (training:256,testing:39),Melanoma(training:288,testing:33),Psoriasis(training:315,testing:35) and Measles (training:279,testing:31).

4.8.2 DEPLOYING THE MODEL

We deployed the model using flask module in python locally. The size of the trained model is around four hundred and eighty megabytes. Due to large size of the model we couldn't upload the project to GitHub, as GitHub supports twenty five mega bytes and larger files requires buying a subscription of github LFS (Large File Storage). Even to deploy in Heroku it requires github integration and file size less than five hundred mega bytes. For these reasons we deployed it on the local machine.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 PERFORMANCE ANALYSIS

We trained the model for 138 epochs until the training accuracy reaches 94% by specifying the condition to stop training when training accuracy reaches 94%>= using callback function in Tensorflow. The validation steps value is set to 25. The time taken for each epoch is around 32 seconds. The testing accuracy couldn't be displayed along with the training accuracy during training period is because to test for epoch the cost of machine resources increases and due to the dearth of availability of resources (GPU) it evaluated at the end (generally this is the industry procedure followed for deep learning systems in contrast with the machine learning models).

The training accuracy of the model is 89.281% accuracy which is considered as the final accuracy of the model. The resolution used for the images is 224x224x3 (3 indicates that the image is coloured(RGB)). The loss function used is CategoricalCrossEntropy since this is a multi class classification. RMSprop with the learning rate of 0.0001 (α) is used as an optimizer. ReLU activation function is used for five dense layers other than pretrained model layers and in the last layer softmax activation function is used.

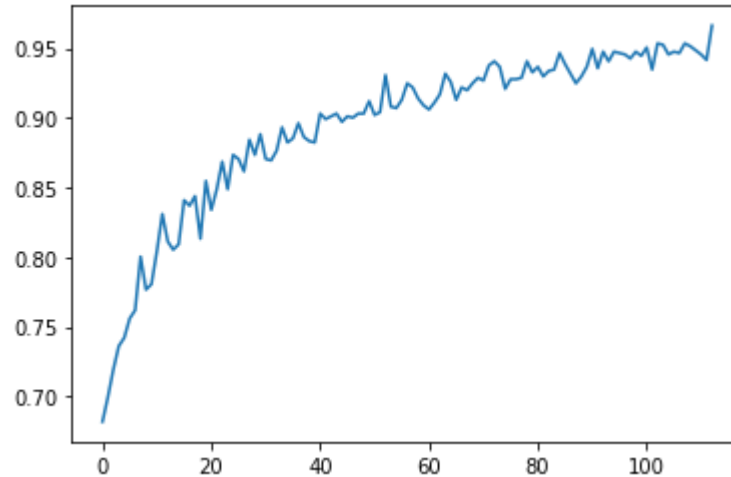


Fig. 5.1. Training accuracy of the model

CHAPTER 6

CONCLUSION

6.1 CONCLUSION

The project explained the method used to detect integumentary diseases by employing Deep learning technology. With the help of transfer learning technique and utilizing Inception neural network, high accuracy can be achieved with a minimal number of images in a dataset. This system cannot replace experts, but it can assist the patient in making a preliminary diagnosis. The model summary is as follows, total parameters: 74,513,668 trainable parameters: 52,710,884 non-trainable parameters: 21,802,784. The prototype created using this method predicted integumentary system disorders with 89.28% accuracy.

REFERENCES

- [1] O. Trabelsi, L. Tlig, M. Sayadi and F. Fnaiech, "Skin disease analysis and tracking based on image segmentation," 2013 International Conference on Electrical Engineering and Software Applications, 2013, pp. 1-7
- [2] A. Ajith, V. Goel, P. Vazirani and M. M. Roja, "Digital dermatology: Skin disease detection model using image processing," 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), 2017, pp. 168-173
- [3] Wirdayanti et al., "Face Skin Disease Detection with Textural Feature Extraction," 2020 6th International Conference on Science in Information Technology (ICSITech), 2020, pp. 133-137
- [4] Z. Ma and J. M. R. S. Tavares, "A Novel Approach to Segment Skin Lesions in Dermoscopic Images Based on a Deformable Model," in IEEE Journal of Biomedical and Health Informatics, vol. 20, no. 2, pp. 615-623, March 2016

- [5] R. Suganya, "An automated computer aided diagnosis of skin lesions detection and classification for dermoscopy images," 2016 International Conference on Recent Trends in Information Technology (ICRTIT), 2016, pp. 1-5

- [6] K. Roy, S. S. Chaudhuri, S. Ghosh, S. K. Dutta, P. Chakraborty and R. Sarkar, "Skin Disease detection based on different Segmentation Techniques," 2019 International Conference on Opto-Electronics and Applied Optics (Optronix), 2019, pp. 1-5

- [7] J. Rathod, V. Waghmode, A. Sodha and P. Bhavathankar, "Diagnosis of skin diseases using Convolutional Neural Networks," 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2018, pp. 1048-1051

- [8] Singh, S. K., & Jalal, A. S. (2016, August). A robust approach for automatic skin cancer disease classification. In 2016 1st India International Conference on Information Processing (IICIP) (pp.1-4). IEEE.

- [9] Z. E. Diame, M. N. Al-Berry, M. A. . -M. Salem and M. Roushdy, "Deep Learning Architectures For Aided Melanoma Skin Disease Recognition: A Review," 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), 2021, pp. 324-329

- [10] X. He et al., "Computer-Aided Clinical Skin Disease Diagnosis Using CNN and Object Detection Models," 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 4839-4844

- [11] Christophers, E. (2001). Psoriasis– epidemiology and clinical spectrum. *Clinical and experimental dermatology*, 26(4), 314-320.

- [12] Bahamdan, K., Mahfouz, A. A., Tallab, T., Badawi, I. A., & AL-AMARI, O. M. (1996). Skin diseases among adolescent boys in Abha, Saudi Arabia. *International journal of dermatology*, 35(6), 405-407
- [13] Karski, T. (2018). Biomechanical Aetiology of the So-called Adolescent Idiopathic Scoliosis (AIS). Lublin Classification (1995-2007). Causative Influences Connected with "Gait" and "Standing 'at ease "on the Right Leg". *J Orthop Bone Res* 1: 102 Abstract Keywords: So-Called Idiopathic Scoliosis.
- [14] Joseph, N., Kumar, G. S., & Nelliyanil, M. (2014). Skin diseases and conditions among students of a medical college in southern India. *Indian dermatology online journal*, 5(1), 19.
- [15] Kolkur, S., & Kalbande, D. R. (2016, November). Survey of texture based feature extraction for skin disease detection. In *2016 International Conference on ICT in Business Industry & Government (ICTBIG)* (pp. 1-6). IEEE
- [16] Chakraborty, S., Mali, K., Chatterjee, S., Anand, S., Basu, A., Banerjee, S., ... & Bhattacharya, A. (2017, October). Image based skin disease detection using hybrid neural network coupled bag-of-features. In *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)* (pp. 242-246). IEEE.
- [17] Kolkur, S., Kalbande, D. R., & Kharkar, V. (2018). Machine learning approaches to multi-class human skin disease detection. *International Journal of Computational Intelligence Research*, 14(1), 1-12.
- [18] Balaji, V. R., Suganthi, S. T., Rajadevi, R., Kumar, V. K., Balaji, B. S., & Pandiyan, S. (2020). Skin disease detection and segmentation using dynamic

graph cut algorithm and classification through Naïve Bayes classifier. Measurement, 163, 107922.

- [19] Chakraborty, S., Mali, K., Chatterjee, S., Banerjee, S., Mazumdar, K. G., Debnath, M., ... & Roy, K. (2017, August). Detection of skin disease using metaheuristic supported artificial neural networks. In 2017 8th Annual Industrial Automation and Electro mechanical Engineering Conference (IEMECON) (pp. 224-229). IEEE.
- [20] Roy, K., Chaudhuri, S. S., Ghosh, S., Dutta, S. K., Chakraborty, P., & Sarkar, R. (2019, March). Skin Disease detection based on different Segmentation Techniques. In 2019 International Conference on Opto-Electronics and Applied Optics (Optronix) (pp. 1-5). IEEE.
- [21] Lu, Y., & Xu, P. (2018). Anomaly detection for skin disease images using variational autoencoder. arXiv preprint arXiv:1807.01349.
- [22] Okuboyejo, D. A., Olugbara, O. O., & Odunaike, S. A. (2013,October). Automating skin disease diagnosis using image classification. In proceedings of the world congress on engineering and computer science (Vol. 2, pp. 850-854).
- [23] Ahmad, B., Usama, M., Huang, C. M., Hwang, K., Hossain, M.S., & Muhammad, G. (2020). Discriminative feature learning for skin disease classification using deep convolutional neural network. IEEE Access, 8, 39025-39033.
- [24] Gu, Y., Ge, Z., Bonnington, C. P., & Zhou, J. (2019). Progressive transfer learning and adversarial domain adaptation for cross-domain skin disease

- classification. IEEE journal of biomedical and health informatics, 24(5), 1379-1393.
- [25] Liao, H., Li, Y., & Luo, J. (2016, December). Skin disease classification versus skin lesion characterization: Achieving robust diagnosis using multi-label deep neural networks. In 2016 23rd International Conference on Pattern Recognition (ICPR) (pp. 355-360). IEEE.
- [26] Hegde, P. R., Shenoy, M. M., & Shekar, B. H. (2018, September). Comparison of machine learning algorithms for skin disease classification using color and texture features. In 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 1825-1828). IEEE.
- [27] Hegde, P. R., Shenoy, M. M., & Shekar, B. H. (2018, September). Comparison of machine learning algorithms for skin disease classification using color and texture features. In 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 1825-1828). IEEE.
- [28] Wei, L. S., Gan, Q., & Ji, T. (2018). Skin disease recognition method based on image color and texture features. Computational and mathematical methods in medicine, 2018.
- [29] Li, W., Zhuang, J., Wang, R., Zhang, J., & Zheng, W. S. (2020, April). Fusing metadata and dermoscopy images for skin disease diagnosis. In 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI) (pp. 1996-2000). IEEE.
- [30] Lopez, A. R., Giro-i-Nieto, X., Burdick, J., & Marques, O. (2017, February). Skin lesion classification from dermoscopic images using deep learning

- techniques. In 2017 13th IASTED international conference on biomedical engineering (BioMed) (pp. 49-54). IEEE.
- [31] Thurnhofer-Hemsi, K., López-Rubio, E., Domínguez, E., & Elizondo, D. A. (2021). Skin lesion classification by ensembles of deep convolutional networks and regularly spaced shifting. *IEEE Access*, 9, 112193-112205.
- [32] F. P. dos Santos and M. A. Ponti, “Robust feature spaces from pre-trained deep network layers for skin lesion classification,” in *Proc. 31st SIBGRAPI Conf. Graph., Patterns Images (SIBGRAPI)*, Oct. 2018, pp. 189–196.
- [33] A. H. Shahin, A. Kamal, and M. A. Elattar, “Deep ensemble learning for skin lesion classification from dermoscopic images,” in *Proc. 9th Cairo Int. Biomed. Eng. Conf. (CIBEC)*, Dec. 2018, pp. 150–153.
- [34] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Proc. NIPS*, 2014, pp. 3320–3328
- [35] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., ... & He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1), 43-76.
- [36] S. Jialin Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
- [37] <https://www.kaggle.com/shubhamgoel27/dermnet>

APPENDICES

A. SOURCE CODE

Index.html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8" />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

```
<meta http-equiv="X-UA-Compatible" content="ie=edge" />
```

```
<title>FunGI!</title>
```

```

<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous" />
<link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.7.2/css/all.css" integrity="sha384-
fNmOCqbTlWlIj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr"
crossorigin="anonymous">
<link href="{{ url_for('static', filename='css/Style.css') }}" rel="stylesheet">
</head>

```

```

<body>
<header>
<div class="container-fluid p-0">
<nav class="navbar navbar-expand-lg">
<a class="navbar-brand" href="#">
<i class="fa fa-shield-virus" aria-hidden="true"></i> <b>FunGI!</b></a>
</nav>
</div>
<div class="container text-center">
<div class="row">
<div class="col-md-7 col-sm-12 text-white">
<h1>Terminate the Fungus</h1>
<p>
We save your money and time and make you feel comfortable all the time
using deep learning.
</p>
<a class="btn btn-light px-5 py-2 primary-btn" href="/second">Let's test</a>
</div>
<div class="col-md-5 col-sm-12 h-25">

```

```

        
    </div>
</div>
</div>
</header>
<main>
    <section class="section-3 container-fluid p-0 text-center">
        <div class="row">
            <div class="col-md-12 col-sm-12">
                <h1>Access Our Application over Multiple Platforms</h1>
                <p>
                    Users from across the globe can access our website through multiple
                    platforms connected to internet. We are working on connecting with users offline.
                </p>
            </div>
        </div>
        <div class="platform row">
            <div class="col-md-6 col-sm-12 text-right">
                <div class="desktop shadow-lg">
                    <div class="d-flex flex-row justify-content-center">
                        <i class="fas fa-desktop fa-3x py-2 pr-3"></i>
                        <div class="text text-left">
                            <h3 class="pt-1 m-0">Desktop</h3>
                            <p class="p-0 m-0">On Website</p>
                        </div>
                    </div>
                </div>
            </div>
            <div class="col-md-6 col-sm-12 text-left">
                <div class="desktop shadow-lg">
                    <div class="d-flex flex-row justify-content-center">

```

```

<i class="fas fa-mobile-alt fa-3x py-2 pr-3"></i>
<div class="text text-left">
  <h3 class="pt-1 m-0">On Mobile</h3>
  <p class="p-0 m-0">On Website</p>
</div>
</div>
</div>
</div>
</div>
</section>

```

```

<!-- Section 4 -->
<section class="section-4">
  <div class="container text-center">
    <h1 class="text-dark">Our Team</h1>
    <p class="text-secondary">We have the best individuals, teamed up to provide
you the best service.</p>
  </div>
  <div class="team row " id="team">
    <div class="col-md-4 col-12 text-center">
      <div class="card mr-2 d-inline-block shadow-lg">
        <div class="card-img-top">
          
        </div>
        <div class="card-body">
          <h3 class="card-title">M. Sai Ram Asish</h3>
          <p class="card-text">
            Researcher and engrossed in the fields of Machine learning, Fullstack-
Web Development, Cyber security.
          </p>

```

```

        <a href="https://www.linkedin.com/in/sai-ram-asish-m/" class="text-
secondary text-decoration-none">Backend Developer</a>
        <p class="text-black-50">Graduate of Sathyabama University,
Chennai</p>
    </div>
</div>
</div>
<div class="col-md-4 col-12 text-center">
    <div class="card mr-2 d-inline-block shadow-lg">
        <div class="card-img-top">
            
        </div>
        <div class="card-body">
            <h3 class="card-title">Jakka Mithil Sai</h3>
            <p class="card-text">
                Researcher and engrossed in the fields of Data analysis, UI/UX Design,
Fullstack-Web Development.
            </p>
            <a href="https://www.linkedin.com/in/mithil1729/" class="text-secondary
text-decoration-none">UI/UX and Frontend Developer</a>
            <p class="text-black-50">Graduate of Sathyabama University,
Chennai</p>
        </div>
    </div>
</div>
</div>
</section>
</main>
<!-- ===== Footer ===== -->
<footer id="footer">

```

```

<div class="container">
  <div class="copyright">
    &copy; Copyright 2022 <strong><span>FunGI</span></strong> . All Rights are
Reserved
  </div>
  <div class="credits">
    Created by <a href="https://www.linkedin.com/in/sai-ram-asish-m/">Sai Ram
Asish</a>, <a href="https://www.linkedin.com/in/mithil-sai-7b5a9116a/">Mithil
Sai</a>
  </div>
</div>
</footer><!-- End Footer -->>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-
UO2eT0CpHqdSJK6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W
1"
crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIlly6OrQ6VrjIEaFf/nJGzlxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
<script src="{{ url_for('static', filename='js/page.js') }}"></script>
</body>
</html>

```

Base.html

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>FunGI</title>
  <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous" />
  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css"
integrity="sha384-
fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr"
crossorigin="anonymous">
  <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
  <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
  <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
  <link href="{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
</head>

<body>
  <header>
    <nav class="navbar navbar-expand-lg">
      <a class="navbar-brand" href="/">
        <i class="fa fa-shield-virus" aria-hidden="true"></i> <b>FunGI!</b></a>
```

```

</nav>
</header>
<div class="container">
  <div id="content" style="margin-top:2em">{% block content %}{% endblock
%}</div>
</div>
<!-- ===== Footer ===== -->
</body>
<footer id="footer">
  <div class="copyright">
    &copy; Copyright 2022 <strong><span>FunGI</span></strong> . All Rights are
Reserved
  </div>
  <div class="credits">
    Created by <a href="https://www.linkedin.com/in/sai-ram-asish-m/">Sai Ram
Asish</a>, <a
    href="https://www.linkedin.com/in/mithil-sai-7b5a9116a/">Mithil Sai</a>
  </div>
</footer><!-- End Footer -->>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-
UO2eT0CpHqdSJJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIlly6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
<script src="{ { url_for('static', filename='js/main.js') } }" type="text/javascript"></script>

```

</html>

Page.html

```
{% extends "base.html" %} {% block content %}
```

```
<div class="rep">
```

```
<a class="btn btn-info px-5 py-2 primary-btn" href="/data">Report</a>
```

```
</div>
```

```
<div class="text-center">
```

```
<h2>Let's Diagnose your problem</h2><br>
```

```
<div>
```

```
<form id="upload-file" method="post" enctype="multipart/form-data">
```

```
<h3>Hello sapien, Kindly upload the infected portion</h3>
```

```
<label for="iname">Picture of the Infection &#8594</label>
```

```
<label for="imageUpload" class="upload-label">
```

```
    Choose
```

```
</label>
```

```
<input type="file" name="file" id="imageUpload" accept=".png, .jpg, .jpeg">
```

```
</form>
```

```
<div class="image-section" style="display:none;">
```

```
<div class="img-preview">
```

```
<div id="imagePreview">
```

```
</div>
```

```
<h3 id="result">
```

```
<span></span>
```

```
</h3>
```

```
</div>
```

```
<div class="mis">
    <button type="submit" class="btn btn-primary btn-lg " id="btn-
predict">Scan</button>
</div>
</div>
```

```
</div>
```

```
</div>
```

```
{% endblock %}
```

```
Data.html
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <title>FunGI</title>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<!--
```

```
=====
```

```
=====-->
```

```
    <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/data.css')
}}">
```

```
<!--
```

```
=====
```

```
=====-->
```

```
</head>
```

```
<body>
```

```
    <div class="contact1">
```

```

<div
class="container-contact1">

<div
class="contact1-pic js-tilt" data-tilt>



</div>

<form
action="/pdf" method="POST" class="contact1-form validate-form">

<span
class="contact1-form-title">

    Kindly Fill the Form for the Report

</span>

<div
class="wrap-input1 validate-input" data-validate = "Name is required">

    <input class="input1" type="text" name="name" id="name" placeholder="Name">

    <span class="shadow-input1"></span>

</div>

<div
class="wrap-input1 validate-input" data-validate = "Valid email is required:
ex@abc.xyz">

    <input class="input1" type="text" name="email" id="email"placeholder="Email">

```

```

        <span class="shadow-input1"></span>
    </div>

    <div
class="wrap-input1 validate-input" data-validate = "Age">

        <input class="input1" type="text" name="age" id="age" placeholder="Age">

        <span class="shadow-input1"></span>
    </div>

    <div
class="container-contact1-form-btn">

        <button class="contact1-form-btn" onclick="handleSubmit()">

            <span>

                Submit

                <i class="fa fa-long-
arrow-right" aria-hidden="true"></i>

            </span>

        </button>
    </div>
</div>
    </div>

```

```
<script >
```

```
$('.js-tilt').tilt({  
    scale: 1.1  
})
```

```
</script>
```

```
<!-- Global site tag (gtag.js) - Google Analytics -->
```

```
<script async src="https://www.googletagmanager.com/gtag/js?id=UA-23581568-13"></script>
```

```
<script>
```

```
    window.dataLayer = window.dataLayer || [];  
    function gtag(){dataLayer.push(arguments);}  
    gtag('js', new Date());
```

```
    gtag('config', 'UA-23581568-13');
```

```
</script>
```

```
<!--
```

```
=====
```

```
<script src="{{ url_for('static', filename='js/data.js') }}"></script>
```

```
</body>
```

```
</html>
```

Pdf.html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>FunGI</title>
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
crossorigin="anonymous">
<link rel="stylesheet" href="{{ url_for('static', filename='css/pdf.css') }}" />
<script src="{{ url_for('static', filename='js/pdf.js') }}"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/html2pdf.js/0.9.2/html2pdf.bundle.js"></script>

</head>

<body>
<div class="container d-flex justify-content-center mt-50 mb-50">
<div class="row">
<div class="col-md-12 text-right mb-3">
<button class="btn btn-primary" id="download">Download</button>
</div>
<div class="col-md-12">
<div class="card" id="invoice">
<div class="card-header bg-transparent header-elements-inline">
<h6 class="card-title text-primary">Diagnose Report - FunGI</h6>
</div>
<div class="card-body">
<div class="row">
<div class="col-sm-6">
<div class="mb-4 pull-left">

<ul class="list list-unstyled mb-0 text-left">
<li>Sathyabama University</li>

```

```

        <li>Chennai</li>
        <li>+91 9087654321</li>
    </ul>
</div>
</div>
<div class="col-sm-6">
    <div class="mb-4 ">
        <div class="text-sm-right">
            <h4 class="invoice-color mb-2 mt-md-2">Disease - <span
id="result-disease" /></h4>
            <ul class="list list-unstyled mb-0">
                <li>Date: <span class="font-weight-semibold">March,
2022</span></li>
            </ul>
        </div>
    </div>
</div>
</div>
<div class="d-md-flex flex-md-wrap">
    <div class="mb-4 mb-md-2 text-left"> <span class="text-
muted">Report To:</span>
        <ul class="list list-unstyled mb-0">
            <li>
                <h5 class="my-2"><span id="result-name" /></h5>
            </li>
            <li><span id="result-age" /></li>
            <li><a href="#" data-abc="true"><p class="address"><span
id="result-email" /></a></li>
        </ul>
    </div>

```

```

</div>
</div>
<div class="table-responsive">
  <table class="table table-lg">
    <thead>
      <tr>
        <th>Description</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>
          <h6 class="mb-0">Do's and Dont's</h6> <span class="text-
muted"><p>1. DO use a mild non-abrasive cleanser<br>
          2. DON'T overdo face washing<br>
          3. DO watch your sun exposure<br>
          4. DON'T be cruel to your skin<br>
          5. DO watch what you eat and drink<br>
          6. DO clean your phone, Yes you read it correctly!<br>
          7. DON'T over examine your skin<br>
          8. DO see a dermatologist<br>
          9. Don't tan. Do use sunscreen<br>
          </p></span>
        </td>
        <td>
          <h6 class="mb-0">General Medication</h6> <span
class="text-muted"><p>
          <p>1. Clotrimazole (Canesten)<br>
          2. Econazole.<br>
          3. Miconazole.<br>
          4. Terbinafine (Lamisil)<br>

```

5. Fluconazole (Diflucan)

6. Ketoconazole (Daktarin)

7. Nystatin (Nystan)

8. Amphotericin.
</p>

</td>

</tr>

</tbody>

</table>

</div>

<div class="card-footer"> The Site cannot and does not contain medical/health advice. The medical/health information is provided for general

informational and educational purposes only and is not a substitute for professional advice. Accordingly, before taking

any actions based upon such information, we encourage you to consult with the appropriate professionals. We do not

provide any kind of medical/health advice. THE USE OR RELIANCE OF ANY INFORMATION CONTAINED ON THE SITE IS SOLELY AT

YOUR OWN RISK. </div>

</div>

</div>

</div>

</div>

</body>

</html>

Main.js

```

$(document).ready(function () {
    // Init
    $('.image-section').hide();
    $('.loader').hide();
    $('#result').hide();
    $('.rep').hide();

    // Upload Preview
    function readURL(input) {
        if (input.files && input.files[0]) {
            var reader = new FileReader();
            reader.onload = function (e) {
                $('#imagePreview').css('background-image', 'url(' + e.target.result + ')');
                $('#imagePreview').hide();
                $('#imagePreview').fadeIn(650);
            }
            reader.readAsDataURL(input.files[0]);
        }
    }
    $("#imageUpload").change(function () {
        $('.image-section').show();
        $('#btn-predict').show();
        $('#result').text("");
        $('#result').hide();
        readURL(this);
    });

    // Predict
    $('#btn-predict').click(function () {
        var form_data = new FormData($('#upload-file')[0]);
    });

```

```

// Show loading animation
$(this).hide();
$('.loader').show();

// Make prediction by calling api /predict
$.ajax({
  type: 'POST',
  url: '/predict',
  data: form_data,
  contentType: false,
  cache: false,
  processData: false,
  async: true,
  success: function (data) {
    // Get and display the result
    $('.loader').hide();
    $('#result').fadeIn(600);
    $('#result').text('You are affected with ' + data);
    $('.rep').show();
    const disease = data;
    localStorage.setItem("DISEASE", disease);
    console.log('Data verified as ' + data);
  },
});

});
Page.js

```

```

let nCount = selector => {
  $(selector).each(function () {
    $(this)
      .animate({
        Counter: $(this).text()
      }, {
        // A string or number determining how long the animation will run.
        duration: 4000,
        // A string indicating which easing function to use for the transition.
        easing: "swing",
        /**
         * A function to be called for each animated property of each animated element.
         * This function provides an opportunity to
         * modify the Tween object to change the value of the property before it is set.
         */
        step: function (value) {
          $(this).text(Math.ceil(value));
        }
      });
  });
};

```

```

let a = 0;
$(window).scroll(function () {
  // The .offset() method allows us to retrieve the current position of an element relative
  to the document
  let oTop = $(".numbers").offset().top - window.innerHeight;
  if (a == 0 && $(window).scrollTop() >= oTop) {
    a++;
    nCount(".rect > h1");
  }
});

```

```
});
```

```
/**
```

```
 *
```

```
 * sticky navigation
```

```
 *
```

```
 */
```

```
let navbar = $(".navbar");
```

```
$(window).scroll(function () {
```

```
  // get the complete hight of window
```

```
  let oTop = $(".section-2").offset().top - window.innerHeight;
```

```
  if ($(window).scrollTop() > oTop) {
```

```
    navbar.addClass("sticky");
```

```
  } else {
```

```
    navbar.removeClass("sticky");
```

```
  }
```

```
});
```

```
(function (root, factory) {
```

```
  if (typeof define === 'function' && define.amd) {
```

```
define([], (function ()
```

```
{
```

```
  return
```

```
factory(root);
```

```
}));
```

```
  } else if (typeof exports === 'object') {
```

```
module.exports =
```

```
factory(root);
```

```

    } else {
        root.SmoothScroll =
factory(root);
    }
})(typeof global !== 'undefined' ? global : typeof window !== 'undefined' ? window : this,
(function (window) {

    'use strict';

    //
    // Default settings
    //

    var defaults = {

        // Selectors
        ignore: '[data-scroll-
ignore]',

        header: null,
        topOnEmptyHash:

        true,

        // Speed & Duration
        speed: 1000,
        speedAsDuration:

        false,

        durationMax: null,
        durationMin: null,
        clip: true,
        offset: 50,

```

```

'easeInOutCubic',

// Easing
easing:

customEasing: null,

// History
updateURL: true,
popstate: true,

// Custom Events
emitEvents: true

};

//
// Utility Methods
//

/**
 * Check if browser supports required methods
 * @return {Boolean} Returns true if all required methods are supported
 */
var supports = function () {

return (

'querySelector' in document &&

'addEventListener' in window &&

'requestAnimationFrame' in window &&

```

```

window.Element.prototype

    'closest' in

    );

};

/**
 * Merge two or more objects together.
 * @param {Object} objects The objects to merge together
 * @returns {Object} Merged values of defaults and options
 */
var extend = function () {
    var merged = {};

    Array.prototype.forEach.call(arguments, (function (obj) {
        for (var key in
        if
        (!obj.hasOwnProperty(key)) return;

        merged[key] = obj[key];

    }
    ));
    return merged;
};

/**
 * Check to see if user prefers reduced motion
 * @param {Object} settings Script settings
 */
var reduceMotion = function () {

```

```

        if ('matchMedia' in
window && window.matchMedia('(prefers-reduced-motion)').matches) {
            return true;
        }
        return false;
    };

    /**
     * Get the height of an element.
     * @param {Node} elem The element to get the height of
     * @return {Number} The element's height in pixels
     */
    var getHeight = function (elem) {
        return
        parseInt(window.getComputedStyle(elem).height, 10);
    };

    /**
     * Escape special characters for use with querySelector
     * @author Mathias Bynens
     * @link https://github.com/mathiasbynens/CSS.escape
     * @param {String} id The anchor ID to escape
     */
    var escapeCharacters = function (id) {

        // Remove leading
        hash

        if (id.charAt(0) ===

        '#' ) {

            id =

            id.substr(1);

```

String(id);

string.length;

string.charCodeAt(0);

length) {

string.charCodeAt(index);

there's no need to special-case astral symbols, surrogate

lone surrogates.

character is NULL (U+0000), then throw an

`InvalidCharacterError` exception and terminate these steps.

=== 0x0000) {

new InvalidCharacterError(

'Invalid character: the input contains U+0000.'

}

var string =

var length =

var index = -1;

var codeUnit;

var result = "";

var firstCodeUnit =

while (++index <

codeUnit =

// Note:

// pairs, or

// If the

//

if (codeUnit

throw

);

```

    }

    if (
        // If the
        character is in the range [\1-\1F] (U+0001 to U+001F) or is
        //
        U+007F, [...]

        (codeUnit >= 0x0001 && codeUnit <= 0x001F) || codeUnit == 0x007F ||
        // If the
        character is the first character and is in the range [0-9]
        //
        (U+0030 to U+0039), [...]

        (index
        === 0 && codeUnit >= 0x0030 && codeUnit <= 0x0039) ||
        // If the
        character is the second character and is in the range [0-9]
        //
        (U+0030 to U+0039) and the first character is a `·` (U+002D), [...]

        (

        index === 1 &&

        codeUnit >= 0x0030 && codeUnit <= 0x0039 &&

        firstCodeUnit === 0x002D

        )

    ){
        //
http://dev.w3.org/csswg/cssom/#escape-a-character-as-code-point

```

	result
+= '\\' + codeUnit.toString(16) + ' ';	
continue;	}
	// If the
character is not handled by one of the above rules and is	
	// greater
than or equal to U+0080, is '-' (U+002D) or '_' (U+005F), or	
	// is in one of
the ranges [0-9] (U+0030 to U+0039), [A-Z] (U+0041 to	
	// U+005A),
or [a-z] (U+0061 to U+007A), [...]	
	if (
codeUnit >= 0x0080	
codeUnit === 0x002D	
codeUnit === 0x005F	
codeUnit >= 0x0030 && codeUnit <= 0x0039	
codeUnit >= 0x0041 && codeUnit <= 0x005A	
codeUnit >= 0x0061 && codeUnit <= 0x007A	) {
	// the
character itself	

```

                                result
+= string.charAt(index);

                                }

                                // Otherwise,

the escaped character.

                                //

http://dev.w3.org/csswg/cssom/#escape-a-character

                                result += '\\' +

string.charAt(index);

                                }

                                // Return sanitized

hash

                                return '#' + result;

};

/**
 * Calculate the easing pattern
 * @link https://gist.github.com/gre/1650294
 * @param {String} type Easing pattern
 * @param {Number} time Time animation should take to complete
 * @returns {Number}
 */
var easingPattern = function (settings, time) {

                                var pattern;

```

```

// Default Easing

Patterns

    if (settings.easing
=== 'easeInQuad') pattern = time * time; // accelerating from zero velocity

    if (settings.easing
=== 'easeOutQuad') pattern = time * (2 - time); // decelerating to zero velocity

    if (settings.easing
=== 'easeInOutQuad') pattern = time < 0.5 ? 2 * time * time : -1 + (4 - 2 * time) * time; //
acceleration until halfway, then deceleration

    if (settings.easing
=== 'easeInCubic') pattern = time * time * time; // accelerating from zero velocity

    if (settings.easing
=== 'easeOutCubic') pattern = (--time) * time * time + 1; // decelerating to zero velocity

    if (settings.easing
=== 'easeInOutCubic') pattern = time < 0.5 ? 4 * time * time * time : (time - 1) * (2 * time -
2) * (2 * time - 2) + 1; // acceleration until halfway, then deceleration

    if (settings.easing
=== 'easeInQuart') pattern = time * time * time * time; // accelerating from zero velocity

    if (settings.easing
=== 'easeOutQuart') pattern = 1 - (--time) * time * time * time; // decelerating to zero
velocity

    if (settings.easing
=== 'easeInOutQuart') pattern = time < 0.5 ? 8 * time * time * time * time : 1 - 8 * (--time)
* time * time * time; // acceleration until halfway, then deceleration

    if (settings.easing
=== 'easeInQuint') pattern = time * time * time * time * time; // accelerating from zero
velocity

    if (settings.easing
=== 'easeOutQuint') pattern = 1 + (--time) * time * time * time * time; // decelerating to
zero velocity

```

```

        if (settings.easing
=== 'easeInOutQuint') pattern = time < 0.5 ? 16 * time * time * time * time * time : 1 + 16
* (--time) * time * time * time * time; // acceleration until halfway, then deceleration

```

```

        // Custom Easing

```

Patterns

```

        if

```

```

        (!!settings.customEasing) pattern = settings.customEasing(time);

```

```

        return pattern ||

```

```

time; // no easing, no acceleration

```

```

    };

```

```

/**

```

```

    * Determine the document's height

```

```

    * @returns {Number}

```

```

    */

```

```

    var getDocumentHeight = function () {

```

```

        return Math.max(

```

```

        document.body.scrollHeight, document.documentElement.scrollHeight,

```

```

        document.body.offsetHeight, document.documentElement.offsetHeight,

```

```

        document.body.clientHeight, document.documentElement.clientHeight

```

```

        );

```

```

    };

```

```

/**

```

```

    * Calculate how far to scroll

```

* Clip support added by robjtede - <https://github.com/cferdinandi/smooth-scroll/issues/405>

* @param {Element} anchor The anchor element to scroll to

* @param {Number} headerHeight Height of a fixed header, if any

* @param {Number} offset Number of pixels by which to offset scroll

* @param {Boolean} clip If true, adjust scroll distance to prevent abrupt stops near the bottom of the page

* @returns {Number}

*/

```
var getEndLocation = function (anchor, headerHeight, offset, clip) {  
    var location = 0;  
    if  
(anchor.offsetParent) {  
        do {  
            location += anchor.offsetTop;  
            anchor = anchor.offsetParent;  
        } while  
(anchor);  
    }  
    location =  
Math.max(location - headerHeight - offset, 0);  
    if (clip) {  
        location =  
Math.min(location, getDocumentHeight() - window.innerHeight);  
    }  
    return location;  
};  
  
/**
```

```

* Get the height of the fixed header
* @param {Node} header The header
* @return {Number} The height of the header
*/
var getHeaderHeight = function (header) {
    return !header ? 0 :
    (getHeight(header) + header.offsetTop);
};

/**
* Calculate the speed to use for the animation
* @param {Number} distance The distance to travel
* @param {Object} settings The plugin settings
* @return {Number} How fast to animate
*/
var getSpeed = function (distance, settings) {
    var speed =
    settings.speedAsDuration ? settings.speed : Math.abs(distance / 1000 * settings.speed);
    if
    (settings.durationMax && speed > settings.durationMax) return settings.durationMax;
    if
    (settings.durationMin && speed < settings.durationMin) return settings.durationMin;
    return
    parseInt(speed, 10);
};

var setHistory = function (options) {

    // Make sure this
    should run

```

```

if
(!history.replaceState || !options.updateURL || history.state) return;

// Get the hash to
use
var hash =
window.location.hash;
hash = hash ? hash
: "";

// Set a default
history
history.replaceState(
{
smoothScroll: JSON.stringify(options),
anchor: hash ? hash : window.pageYOffset
},
document.title,
hash ? hash :
window.location.href
);

};

/**
 * Update the URL
 * @param {Node} anchor The anchor that was scrolled to

```

```

* @param {Boolean} isNum If true, anchor is a number
* @param {Object} options Settings for Smooth Scroll
*/
var updateURL = function (anchor, isNum, options) {

// Bail if the anchor
is a number

if (isNum) return;

// Verify that
pushState is supported and the updateURL option is enabled
if
(!history.pushState || !options.updateURL) return;

// Update URL
history.pushState(
    {
smoothScroll: JSON.stringify(options),

anchor: anchor.id

    },

document.title,

anchor ===

document.documentElement ? '#top' : '#' + anchor.id

);

};

/**

```

```

* Bring the anchored element into focus
* @param {Node}  anchor    The anchor element
* @param {Number} endLocation The end location to scroll to
* @param {Boolean} isNum    If true, scroll is to a position rather than an element
*/
var adjustFocus = function (anchor, endLocation, isNum) {

                                                                    // Is scrolling to top
of page, blur

                                                                    if (anchor === 0) {

document.body.focus();

                                                                    }

                                                                    // Don't run if
scrolling to a number on the page

                                                                    if (isNum) return;

                                                                    // Otherwise, bring
anchor element into focus

                                                                    anchor.focus();
                                                                    if
(document.activeElement !== anchor) {

anchor.setAttribute('tabindex', '-1');

anchor.focus();

anchor.style.outline = 'none';

                                                                    }

```

```

        window.scrollTo(0 ,
endLocation);

};

/**
 * Emit a custom event
 * @param {String} type The event type
 * @param {Object} options The settings object
 * @param {Node} anchor The anchor element
 * @param {Node} toggle The toggle element
 */
var emitEvent = function (type, options, anchor, toggle) {
    if
(!options.emitEvents || typeof window.CustomEvent !== 'function') return;
    var event = new
CustomEvent(type, {
    bubbles: true,
    detail: {
        anchor: anchor,
        toggle:
toggle
    }
});

    document.dispatchEvent(event);
};

//

```

```

// SmoothScroll Constructor
//

var SmoothScroll = function (selector, options) {

//
// Variables
//

var smoothScroll =

var settings, anchor,

toggle, fixedHeader, eventTimeout, animationInterval;

//
// Methods
//

/**
 * Cancel a scroll-in-
 */

progress

smoothScroll.cancelScroll = function (noEvent) {

cancelAnimationFrame(animationInterval);

animationInterval = null;

if (noEvent)

return;

```

```

emitEvent('scrollCancel', settings);

};

/**
 * Start/stop the
scrolling animation
 * @param
{Node|Number} anchor The element or position to scroll to
 * @param
{Element} toggle The element that toggled the scroll event
 * @param {Object}
options
 */

smoothScroll.animateScroll = function (anchor, toggle, options) {

// Cancel any
in progress scrolls

smoothScroll.cancelScroll();

// Local
settings

var _settings
= extend(settings || defaults, options || {}); // Merge user options with defaults

// Selectors
and variables

var isNum =
Object.prototype.toString.call(anchor) === '[object Number]' ? true : false;

```

```

var
anchorElem = isNum || !anchor.tagName ? null : anchor;

if (!isNum &&
!anchorElem) return;

var
startLocation = window.pageYOffset; // Current location on the page

if
(_settings.header && !fixedHeader) {

// Get
the fixed header if not already set

    fixedHeader = document.querySelector(_settings.header);

}
var
headerHeight = getHeaderHeight(fixedHeader);

var
endLocation = isNum ? anchor : getEndLocation(anchorElem, headerHeight,
parseInt((typeof _settings.offset === 'function' ? _settings.offset(anchor, toggle) :
_settings.offset), 10), _settings.clip); // Location to scroll to

var distance
= endLocation - startLocation; // distance to travel

var
documentHeight = getDocumentHeight();

var
timeLapsed = 0;

var speed =
getSpeed(distance, _settings);

var start,
percentage, position;

/**

```

```

* Stop the
scroll animation when it reaches its target (or the bottom/top of page)

* @param
{Number} position Current position on the page

* @param
{Number} endLocation Scroll to location

* @param
{Number} animationInterval How much to scroll on this loop

*/
var
stopAnimateScroll = function (position, endLocation) {

// Get
the current location

var
currentLocation = window.pageYOffset;

//
Check if the end location has been reached yet (or we've hit the end of the document)
if
(position == endLocation || currentLocation == endLocation || ((startLocation <
endLocation && window.innerHeight + currentLocation) >= documentHeight)) {

// Clear the animation timer

smoothScroll.cancelScroll(true);

// Bring the anchored element into focus

```

```

adjustFocus(anchor, endLocation, isNum);

// Emit a custom event

emitEvent('scrollStop', _settings, anchor, toggle);

// Reset start

start = null;

animationInterval = null;

return true;

    }
};

/**
 * Loop
 */
var
scrolling animation

loopAnimateScroll = function (timestamp) {

    if

    (!start) { start = timestamp; }

    timeLapsed += timestamp - start;

```

```

percentage = speed === 0 ? 0 : (timeLapsed / speed);

percentage = (percentage > 1) ? 1 : percentage;

position = startLocation + (distance * easingPattern(_settings, percentage));

window.scrollTo(0, Math.floor(position));

                                                                    if
(!stopAnimateScroll(position, endLocation)) {

    animationInterval = window.requestAnimationFrame(loopAnimateScroll);

    start = timestamp;

                                                                    }
                                                                    };

                                                                    /**
                                                                    * Reset
                                                                    * @link
                                                                    */
                                                                    if
position to fix weird iOS bug

https://github.com/cferdinandi/smooth-scroll/issues/45

                                                                    */
                                                                    if
(window.pageYOffset === 0) {

    window.scrollTo(0, 0);

                                                                    }

                                                                    // Update the
URL

```

```

        updateURL(anchor, isNum, _settings);

                                                                    // If the user
        prefers reduced motion, jump to location
                                                                    if
        (reduceMotion()) {
                                                                    return;
                                                                    }

                                                                    // Emit a
        custom event

        emitEvent('scrollStart', _settings, anchor, toggle);

                                                                    // Start
        scrolling animation

        smoothScroll.cancelScroll(true);

        window.requestAnimationFrame(loopAnimateScroll);

                                                                    };

                                                                    /**
                                                                    * If smooth scroll

        element clicked, animate scroll

                                                                    */

```

```

function (event) {
    // Don't run if
    event was canceled but still bubbled up
    // By
    @mgreter - https://github.com/cferdinandi/smooth-scroll/pull/462/
    if
    (event.defaultPrevented) return;

    // Don't run if
    right-click or command/control + click or shift + click
    if
    (event.button !== 0 || event.metaKey || event.ctrlKey || event.shiftKey) return;

    // Check if
    event.target has closest() method
    // By @totege
    - https://github.com/cferdinandi/smooth-scroll/pull/401/
    if (!('closest'
    in event.target)) return;

    // Check if a
    smooth scroll link was clicked
    toggle =
    event.target.closest(selector);
    if (!toggle ||
    toggle.tagName.toLowerCase() !== 'a' || event.target.closest(settings.ignore)) return;

    // Only run if
    link is an anchor and points to the current page

```

	if
(toggle.hostname !== window.location.hostname toggle.pathname !==	
window.location.pathname !#!/.test(toggle.href)) return;	
	// Get an
escaped version of the hash	
	var hash;
	try {
	hash =
escapeCharacters(decodeURIComponent(toggle.hash));	
	} catch(e) {
	hash =
escapeCharacters(toggle.hash);	
	}
console.log(hash);	
	// Get the
anchored element	
	var anchor;
	if (hash ===
'#') {	
	if
(!settings.topOnEmptyHash) return;	
	anchor
= document.documentElement;	
	} else {
	anchor
= document.querySelector(hash);	
	}

```

anchor =
!anchor && hash === '#top' ? document.documentElement : anchor;

// If anchored
element exists, scroll to it
if (!anchor)
return;

event.preventDefault();

setHistory(settings);

smoothScroll.animateScroll(anchor, toggle);

};

/**
 * Animate scroll on
 */
var popstateHandler

= function (event) {

// Stop if
history.state doesn't exist (ex. if clicking on a broken anchor link).

// fixes
`Cannot read property 'smoothScroll' of null` error getting thrown.
if
(history.state === null) return;

```

```

// Only run if
state is a popstate record for this instantiation
if
(!history.state.smoothScroll || history.state.smoothScroll !== JSON.stringify(settings))
return;

// Only run if
state includes an anchor

// if
(!history.state.anchor && history.state.anchor !== 0) return;

// Get the
anchor
var anchor =
history.state.anchor;
if (typeof
anchor === 'string' && anchor) {
anchor
= document.querySelector(escapeCharacters(history.state.anchor));
if
(!anchor) return;
}

// Animate
scroll to anchor link

smoothScroll.animateScroll(anchor, null, {updateURL: false});

};

```

```

current initialization.
    /**
     * Destroy the
    */

    smoothScroll.destroy = function () {

        // If plugin
        isn't already initialized, stop
        if (!settings)
            return;

        // Remove
        event listeners

        document.removeEventListener('click', clickHandler, false);

        window.removeEventListener('popstate', popstateHandler, false);

        // Cancel any
        scrolls-in-progress

        smoothScroll.cancelScroll();

        // Reset
        variables

        settings =

        null;

        anchor = null;
        toggle = null;

```

```

fixedHeader
= null;

eventTimeout
= null;

animationInterval = null;

};

/**
 * Initialize Smooth
 * @param {Object}
 */
var init = function ()

{

// feature test
if
(!supports()) throw 'Smooth Scroll: This browser does not support the required
JavaScript methods and browser APIs.';

// Destroy any
existing initializations

smoothScroll.destroy();

// Selectors
and variables

```

```

                                                                    settings =
extend(defaults, options || {}); // Merge user options with defaults
                                                                    fixedHeader
= settings.header ? document.querySelector(settings.header) : null; // Get the fixed
header

                                                                    // When a
toggle is clicked, run the click handler

    document.addEventListener('click', clickHandler, false);

                                                                    // If
updateURL and popState are enabled, listen for pop events

                                                                    if
(settings.updateURL && settings.popstate) {

    window.addEventListener('popstate', popstateHandler, false);

                                                                    }

                                                                    };

                                                                    //
                                                                    // Initialize plugin
                                                                    //

                                                                    init();

                                                                    //
                                                                    // Public APIs

```

```

//

return smoothScroll;

};

return SmoothScroll;

));
Data.js
function handleSubmit () {
    const name = document.getElementById('name').value;
    const email = document.getElementById('email').value;
    const age = document.getElementById('age').value;

    localStorage.setItem("NAME", name);
    localStorage.setItem("EMAIL", email);
    localStorage.setItem("AGE", age);

    return;
}
(function ($) {
    "use strict";

    /*=====
    =
    [ Validate ]*/
    var name = $(''.validate-input input[name="name"]);

```

```

var email = $('#validate-input input[name="email"]');
var subject = $('#validate-input input[name="subject"]');
var message = $('#validate-input textarea[name="message"]');

```

```

$('#validate-form').on('submit',function(){
    var check = true;

```

```

    if($('#name').val().trim() == ""){
        showValidate(name);
        check=false;
    }

```

```

    if($('#subject').val().trim() == ""){
        showValidate(subject);
        check=false;
    }

```

```

    if($('#email').val().trim().match(/^[a-zA-Z0-9_\-\.]+@((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. )|([a-zA-Z0-9\-\.] +))([a-zA-Z]{1,5}|[0-9]{1,3})(\?)$/) == null) {
        showValidate(email);
        check=false;
    }

```

```

    if($('#message').val().trim() == ""){
        showValidate(message);
        check=false;
    }

```

```

    return check;

```

```
});
```

```
$('.validate-form .input1').each(function(){  
    $(this).focus(function(){  
        hideValidate(this);  
    });  
});
```

```
function showValidate(input) {  
    var thisAlert = $(input).parent();  
  
    $(thisAlert).addClass('alert-validate');  
}
```

```
function hideValidate(input) {  
    var thisAlert = $(input).parent();  
  
    $(thisAlert).removeClass('alert-validate');  
}  
})(jQuery);
```

Pdf.js

```
window.addEventListener('load', () => {  
  
    const name = localStorage.getItem('NAME');  
    const age = localStorage.getItem('AGE');  
    const email = localStorage.getItem('EMAIL');  
    const disease = localStorage.getItem('DISEASE');
```

```

document.getElementById('result-name').innerHTML = name;
document.getElementById('result-age').innerHTML = age;
document.getElementById('result-email').innerHTML = email;
document.getElementById('result-disease').innerHTML = disease;

```

```

})

```

```

window.onload = function () {
    document.getElementById("download")
        .addEventListener("click", () => {
            const invoice = this.document.getElementById("invoice");
            console.log(invoice);
            console.log(window);
            var opt = {
                margin: 1,
                filename: 'Diagnose Report.pdf',
                image: { type: 'jpeg', quality: 0.98 },
                html2canvas: { scale: 2 },
                jsPDF: { unit: 'in', format: 'tabloid', orientation: 'portrait' }
            };
            html2pdf().from(invoice).set(opt).save();
        })
}

```

App.py

```

from __future__ import division, print_function

```

```

# coding=utf-8
import sys
import os
import glob
import re
import numpy as np
import tensorflow as tf
from PIL import Image, ImageEnhance

from tensorflow.keras.models import load_model
# Keras
from tensorflow.keras.applications.imagenet_utils import preprocess_input,
decode_predictions
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

# Flask utils
from flask import Flask, redirect, url_for, request, render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer
app = Flask(__name__)
MODEL_PATH = 'skin_model2.h5'
model = load_model(MODEL_PATH)
print('Model loaded. Check http://127.0.0.1:5000/')

def softmax(x):
    f_x = np.exp(x) / np.sum(np.exp(x))
    return f_x
class_list=['Psoriasis','Measles','Melanoma','Ringworm']

def predicting(fp, model):

```

```

img=Image.open(fp)
target_size=(224,224)

if img.size != target_size:
    img = img.resize(target_size)
enhancer = ImageEnhance.Contrast(img)
factor=2.5
img = enhancer.enhance(factor)
x = tf.keras.utils.img_to_array(img)
x = np.expand_dims(x, axis=0)
x = tf.keras.applications.inception_v3.preprocess_input(x)
preds = model.predict(x)
tem_ar=softmax(preds[0])
return class_list[tem_ar.argmax()]

```

```

@app.route('/', methods=['GET'])
def index():
    # Main page
    return render_template('index.html')
@app.route('/second', methods=['GET', 'POST'])
def second():
    return render_template('page.html')
@app.route('/data', methods=['GET', 'POST'])
def data():
    return render_template('data.html')
@app.route('/pdf', methods=['GET', 'POST'])
def pdf():
    return render_template('pdf.html')

@app.route('/predict', methods=['GET', 'POST'])
def upload():

```

```

if request.method == 'POST':
    # Get the file from post request
    f = request.files['file']

    # Save the file to ./uploads
    basepath = os.path.dirname(__file__)
    file_path = os.path.join(
        basepath, 'uploads', secure_filename(f.filename))
    f.save(file_path)

    # Make prediction
    result = predicting(file_path, model)

    return result

return None

```

```

if __name__ == '__main__':
    app.run(debug=True)

```

B. SCREEN SHOTS

```

C:\FYP\FunGI>python app.py
2022-03-25 16:32:45.223266: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlderror: cudart64_110.dll not found
2022-03-25 16:32:45.223775: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2022-03-25 16:35:04.078976: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.dll'; dlderror: nvcuda.dll not found
2022-03-25 16:35:04.079184: W tensorflow/stream_executor/cuda/cuda_driver.cc:269] failed call to cuInit: UNKNOWN ERROR (303)
2022-03-25 16:35:04.080450: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: Laptop-MI
2022-03-25 16:35:04.088702: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: Laptop-MI
2022-03-25 16:35:04.123330: I tensorflow/core/platform/cpu_feature_guard.cc:151] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in perf
ormance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Model loaded. Check http://127.0.0.1:5000/
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
2022-03-25 16:35:20.484739: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlderror: cudart64_110.dll not found
2022-03-25 16:35:20.484953: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2022-03-25 16:35:23.557567: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.dll'; dlderror: nvcuda.dll not found
2022-03-25 16:35:23.557762: W tensorflow/stream_executor/cuda/cuda_driver.cc:269] failed call to cuInit: UNKNOWN ERROR (303)
2022-03-25 16:35:23.567599: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: Laptop-MI
2022-03-25 16:35:23.567861: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: Laptop-MI
2022-03-25 16:35:23.568437: I tensorflow/core/platform/cpu_feature_guard.cc:151] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in perf
ormance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Model loaded. Check http://127.0.0.1:5000/
* Debugger is active!
* Debugger PID: 119-140-757
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

```

Fig 1: Flask booting up

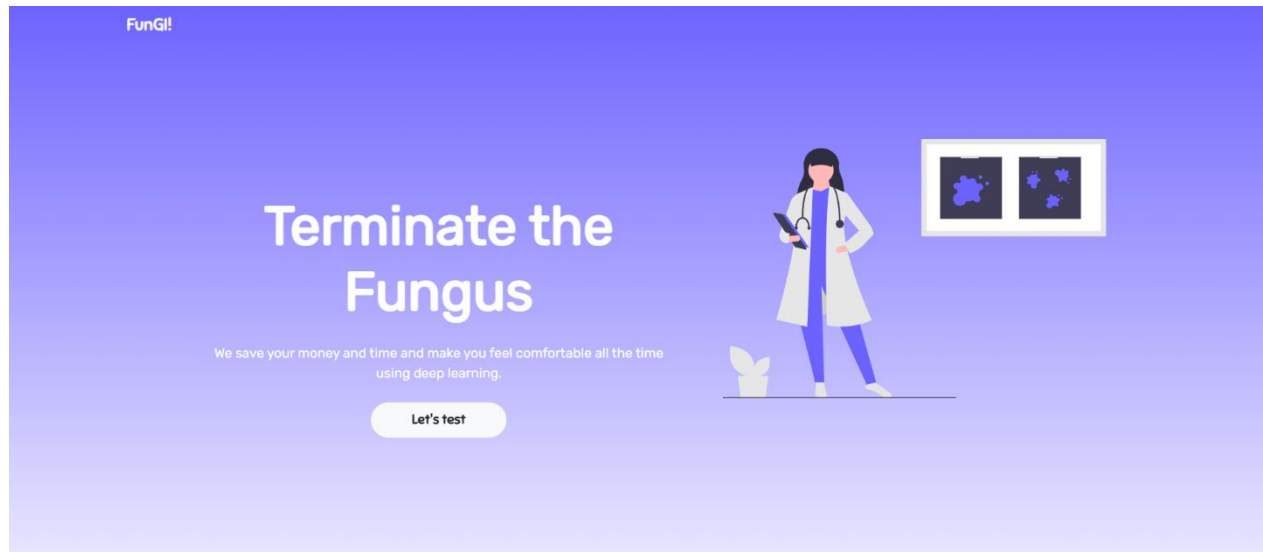


Fig 2: Landing page



Fig 3: Infection area

FunGii!

Report

Let's Diagnose your problem

Hello sapien, Kindly upload the infected portion

Picture of the Infection → Choose



You are affected with Ringworm

Fig 4: Diagnose page

Kindly Fill the Form for the Report



Suresh

suresh@gmail.com

24

Submit

Fig 5: Data insertion of the victim

[Download](#)

Diagnose Report - FunGI

Sathyabama University
Chennai
+91 9087654321

Report To:
Suresh
24
suresh@gmail.com

Disease - Ringworm
Date: March, 2022

Description

Do's and Dont's

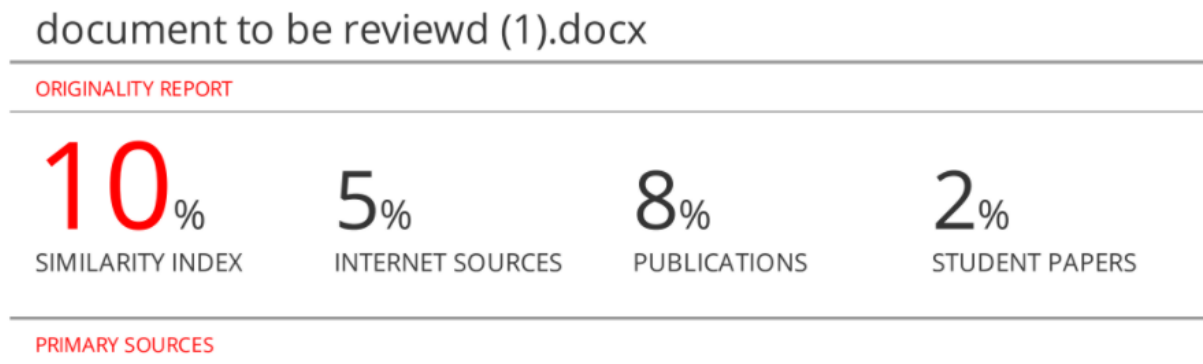
1. DO use a mild non-abrasive cleanser
2. DON'T overdo face washing
3. DO watch your sun exposure
4. DON'T be cruel to your skin
5. DO watch what you eat and drink
6. DO clean your phone, Yes you read it correctly!
7. DON'T over examine your skin
8. DO see a dermatologist
9. Don't tan. Do use sunscreen

General Medication

1. Clotrimazole (Canesten)
2. Econazole.
3. Miconazole.
4. Terbinafine (Lamisil)
5. Fluconazole (Diflucan)
6. Ketoconazole (Daktarin)
7. Nystatin (Nystan)
8. Amphotericin.

Fig 6: Report

C. PLAGIARISM REPORT



D. Journal Paper



6th International Conference on
Trends in Electronics and Informatics
(ICOEI 2022)

28-30, April 2022 | icoei.com/2022 | icoei.conf@gmail.com

Letter of Acceptance

To

Madiraju, Sai ram nish, Jakka, Mithil Sai, R. Aishwarya, Yogitha, R

Department of Computer Science and Engineering Sathyabama Institute of Science and Technology,
Chennai, India. Department of Computer Science and Engineering, Assistant Professor Sathyabama
Institute of Science and Technology, Chennai, India.

Greetings!!

On behalf of the SCAD College of Engineering and Technology Tirunelveli, we take great
pleasure in inviting you to attend the "6th International Conference on Trends in Electronics
and Informatics (ICOEI 2022)" which is going to be held during **28-30, April 2022** at
Tirunelveli, India.

We are glad to inform you that your submitted article entitled "**Categorization of
Integumentary System Disorders using Deep Learning**" [**Paper ID: ICOEI-850**] has been
accepted for publication at ICOEI 2022. We welcome you to join us and share your research and
views. This conference will offer you an unforgettable experience in exploring new opportunities.

With Thanks,

Yours' Sincerely

Dr. R. Karthik Ganesha,
Organizing Secretary
ICOEI 2022.

Previous Publications

ICOEI 2021
ICOEI 2020
ICOEI 2019
ICOEI 2018
ICOEI 2017

Proceedings by



Categorization of Integumentary System Disorders using Deep Learning

Madiraju. Sai ram asish
Department of Computer
Science and Engineering
Sathyabama Institute of Science
and Technology

Chennai, India

Jakka.Mithil Sai
Department of Computer
Science and Engineering
Sathyabama Institute of Science
and Technology

Chennai, India

R. Aishwarya
Department of Computer Science
and Engineering
Assistant Professor Sathyabama
Institute of Science and
Technology

Yogitha.R
Department of Computer
Science and Engineering
Assistant Professor Sathyabama
Institute of Science and
Technology

Abstract— People in today's world are busy and occupied all of the time. We often overlook minor illnesses in our bodies as a result of our fast-paced lives. Skin disease is one of them. It is the most widespread disease on the planet. People generally consider dermatology problems as ephemeral, though that's not always the case. If the skin disease is not appropriately recognized, it might cause serious complications. The project's current models use segmentation techniques such as edge detection; however, our study seeks to diagnose skin disorders with high accuracy rates utilizing Deep learning. Our goal is to use CNN (convolutional neural networks) and transfer learning from the Inception model to identify and categorize skin disorders. This research seeks to inform victims of the potential repercussions and assist doctors in making an initial diagnosis. This is being implemented in both a web application and an Android application.

Keywords— Integumentary system, Deep learning, Diagnosing integumentary system diseases, CNN, Image classification, Computer Vision, Transfer Learning.

INTRODUCTION

AI is a broad term for intelligence exhibited by machines; we are witnessing a wide-reaching integration of AI into human life in the next few decades. Digital assistants, for example, help us with day-to-day tasks and give us information when we need it. Web searching is becoming increasingly accurate and efficient thanks to AI's predictive capabilities; an example is AI-powered web search. It has increased and can now give you accurate answers to your questions, which would have taken hours of research before. The disease detection or natural catastrophe forecasts are far more accurate than they were twenty years ago. With the massive amount of data generated every day, it is natural that AI plays an essential role in the future of daily life. Artificial intelligence has been a topic of discussion for a long time, and it seems that it is finally reaching a point in which it can impact our day-to-day lives. Deep learning is a subdivision of AI. It aims at developing

algorithms that analyze vast datasets to learn how to perform specific jobs.

Deep learning involves neural networks made up of multiple layers. These networks mimic the behaviour of the brain. Neural networks are at the core, and they provide the best results with unstructured data. This form of AI allows for more accessible training and better returns on investment. There are many advantages; one advantage is the high accuracy it provides with unstructured data. Deep learning outperforms machine learning when it comes to addressing complicated problems. The deep learning models' accuracy is high, and their predictions are close to humans.

Most people do not know how to take care of it when it comes to skin. It is a sensitive and delicate organ that needs care and affection. An estimated 1 in 3 people have a skin disease at any time. The human body has many potential entry points for parasites and bacteria to get inside and infect us with various ailments. Without a proper diagnosis, the disease will likely worsen. Some common skin diseases are eczema, psoriasis, acne, rosacea, and ichthyosis.

Classifying integumentary system disorders using deep learning is one of the most challenging tasks for medical professionals. The accuracy of a computer-based diagnosis for a patient can help save time and money by providing a more accurate diagnosis. An important step in diagnosing a medical condition is to classify it as a certain type of disorder. This classification may be done using, for example, the ICD codes or the DSM-5 codes. To classify the group of International Classification of Diseases (ICD)

codes into a suitable category, we use deep learning approach. Our results show that our method outperforms current methods in terms of both accuracy and efficiency.

Our goal is to classify integumentary system disorders namely Acne, Psoriasis, Ringworm and Eczema using Deep learning. The model is trained using images of each disease as input. The images are fed to the deep learning layers, which adjust the weights of the neurons for every epoch. This model uses an Inception v3 neural network with 48 layers and an additional five dense layers for training photos of distinct classes. After completion of training the testing phase carried out to determine the accuracy of the model.

LITERATURE SURVEY

Integumentary system disorders affect 71 percent of the world's population, almost three-fourths of humankind. Ailments of the integumentary system are the third most prevalent illness, and many who suffer from them do not consider them a significant problem that requires medical attention [2].

Depending on the skin state, the appearance of skin conditions in dermoscopy images might differ significantly. Hair, texture, and bubbles may further obfuscate the distinction between skin infections and neighbouring clear skin. As a result, dermoscopy image classification is a complicated process [4]. We are deploying cutting-edge deep learning technologies to take on this complex task.

When compared towards other ml algorithms, deep learning has been used to perform exceptionally challenging classification and segmentation jobs [32], [33] with great accuracy. The framework of the system is based primarily on convolutional layers [31].

Characteristics in the initial layers of deep learning network topologies are typically broad, while high-level features are often particular. Yosinski et al. [34] demonstrated that using model parameters transmitted from different jobs to initialize

networks enhances results compared to arbitrary parameters [24].

[2] Digital dermatology: Skin disease detection model using image processing, this paper uses algorithms like DCT (Direct Cosine Transform), SVD (Single Value Decomposition) and DWT (Direct Wavelet Transform) According to this paper, the attained accuracy for detecting skin disorders is 80%. The model's reliability would have increased greatly if neural networks had been utilized to train the images.

[7] Diagnosis of skin diseases using Convolutional Neural Networks, this paper solely relies on convolutional neural network (CNN) and achieved accuracy of seventy percent. The paper claims that accuracy can be improved if larger dataset is used. The model's accuracy would have improved significantly if the transfer learning approach had been used. Our model was also trained with around one thousand two hundred pictures of skin diseases and tested with two hundred images but achieved around ninety percent accuracy.

[3] Face Skin Disease Detection with Textural Feature Extraction, this paper aims at classifying skin face skin disease detection using K-nearest neighbour algorithm which produced an accuracy of eighty percent. If Deep learning is utilized and transfer learning is implemented then the efficiency of the model would have enhanced greatly.

PROPOSED SYSTEM

The proposed system seeks to develop a deep learning model to identify integumentary system disorders accurately. The prototype is made available in both web browsers and mobile applications. The proliferation of smartphones has made cameras readily available to everyone; this makes the disease diagnosis more interactive and straightforward. The system is not intended to replace a doctor's participation in the diagnosis completely; rather, it is intended to provide a preliminary diagnosis to understand the condition better. Uploading a picture

through a generic camera with ample light is sufficient for recognizing the disease. Fig 1,2,3,4 are the diseases trained in this model. The reason to choose these diseases is that they are widespread among people [14].



Fig. 1. Eczema



Fig. 2. Ringworm



Fig. 3. Acne



Fig. 4. Psoriasis

Model

The dataset was curated from the pictures of the disease on the internet. The model is trained using around three thousand samples. To augment the size of the dataset, we are using the ImageDataGenerator method in TensorFlow framework, which feeds the images in to the model by rotating, shifting the width and height, shear transforming, magnifying, flipping and varying the brightness of the image for every epoch. To further improve the model's accuracy, we are increasing the contrast of the image, which enhances the colours of the image. Any image resolution is tolerable since it will be converted to the standard size for model processing. The model requires inputting age, area of infection, sex to improve the model results further. The enhanced image is passed on to the trained model to receive the output of the disease name. The model predicts the name of the disease with 90.28% accuracy.

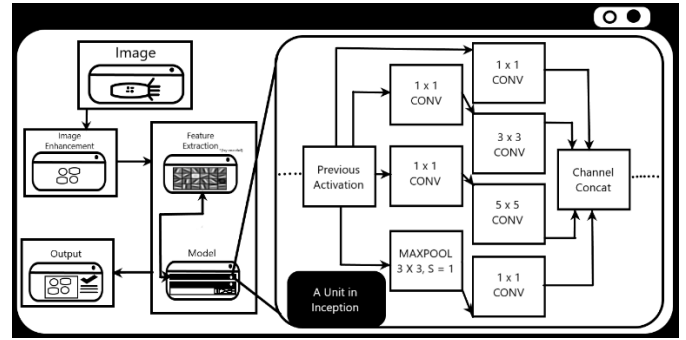


Fig. 5. System Architecture

IV. METHODOLOGY

Though other image classification methods are available, like SVM, we use CNN (Convolutional neural networks) because it accurately picks the features from the image through filters and efficiently compares the adjacent pixel. CNN also automatically learns the features of an image which is not the case with other algorithms. The dataset is initially divided into test, train and validation datasets. The train dataset is used to train the model, i.e., it helps the model to understand and recognize the pattern. The test dataset is used to run the trained model to predict the accuracy. The validation dataset is utilized to fine-tune the model's hyperparameters, such as learning rate. The model employs transfer learning method to predict the integumentary system diseases. The advantage of this method is that it improves the accuracy of the model compared to a model trained from scratch. Out of all the neural network models, the proposed system uses the Inception network developed by Google. The Inception network is trained with a million pictures and the number of convolutional layers are 48.

A. Convolutions

Convolution is a convolutional neural network method that includes the product of a weight matrix with the inputs. The multiplication is done between input data and 2D matrix of values, named filter.

The dot product is used to multiply the corresponding size of the filter in the input with the filter. dot product is the element-wise product of the input and filter's area, always yielding a specific value when totaled. Different types of filters are available as per our requirement, like edge detection filter and

sharpen image filter. These filters also have a bias factor that can adjust the bias in a specific layer.as shown in Fig. 6.

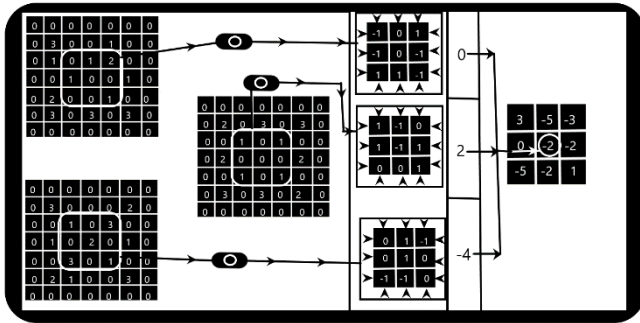


Fig. 6. Working of convolutional layers

The general formula for calculating the output volume involves V is input volume size, F is kernel size, S is stride length, P being padding.

$$\frac{V - F + 2P}{S} + 1$$

Convolutional networks use an activation function, ReLU (Rectified Linear Activation Unit) function is generally used in most neural networks since it offers excellent performance and is easier to train the model. The general equation of ReLU is

$$Re(Z) = \max(0, Z)$$

The ReLU is a widely used activation function which doesn't change the value of the input if it is greater than zero and if the input value is less than zero it outputs the value as zero. The computation of cost for using this activation function is very low in comparison with other activation functions like sigmoid, tanh, etc. Plotting the graph for the below function where x is the input and f(x) is the activation (ReLU) can be observed in Fig.9.

$$f(x) = \begin{cases} 0, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases}$$

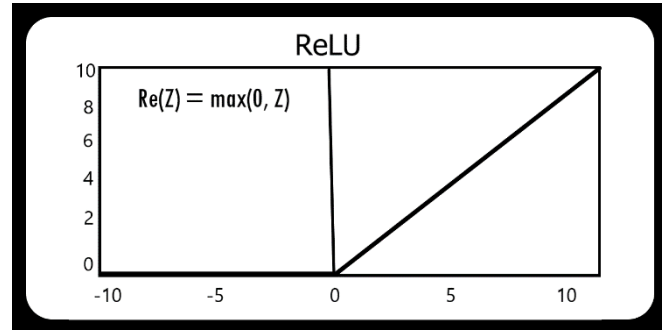


Fig. 7. ReLU graph

Pooling is a technique used to reduce the image size, for example, decreasing from 8x8 image to 4x4 image. As depicted in Fig.10, Max pooling is employed in the Inception network. Based on the stride length, the maximum value in the patch of input image is considered.

$$f_{(u,v)}(I) = \max_{m,n=0}^1 I_{(2U+m, 2V+n)}$$

In the above equation I is the matrix, U, V are the axes with 2x2 filter and number of pixels shifted also called as stride length is 2. The max of 2x2 filter with stride length of 2 is applied as shown in Fig. 10.

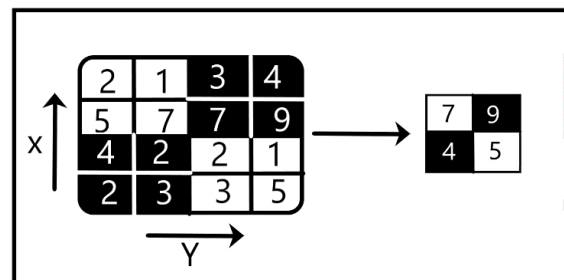


Fig. 8. Max pooling.

1x1 convolution layers are used in the Inception network to decrease the complexity i.e., they are capable of reducing CNN output layers, for example, 8x8x22 can be reduced to 8x8x10, as shown in Fig. 8.

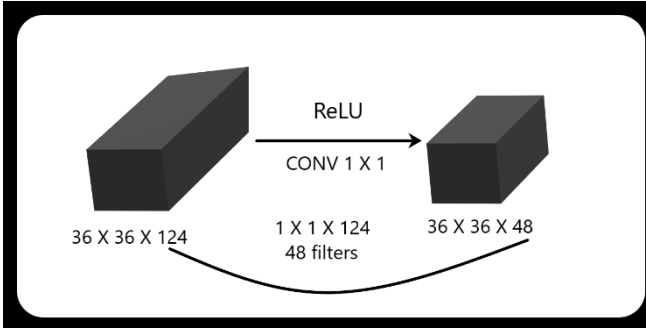


Fig. 9. 1x1 convolution

Transfer Learning

The number of images required to train the model is drastically reduced using transfer learning technology. Generally, transfer learning is used when large dataset of pictures are not available. It also helps save the system's resources and does not compromise accuracy. In this paper, we use Inception network v3 though many architectures like VGG and Alexnet are available.

Formal learning is unique to each project and we need to train the model for a specific task. There is no provision for transferring the acquired knowledge to another model. This is a huge drawback since training models again and again for similar projects takes up many resources. Transfer learning enables us to use pre-trained models' information to train new models and even solves challenges like insufficient data for the newer model.

The components used in Inception neural network consists of 1x1 convolution, Max pooling and convolution filters with 3x3 and 5x5 filters which can be observed in Fig. 9,8,6. The architecture of inception can be observed in the Fig.4.

Neural networks are designs with many hyperparameters that may be customized. The first layers capture general information and the following layers concentrate on the individual job. When training again, we can lock specific layers or use fine-tuning for the remainder to fulfill

the requirements. With careful study of the architecture, we can use instances of the Inception network as our starting point of the model. With this feature of transfer learning, we can get the best efficiency in minimal time, as depicted in Fig. 11.

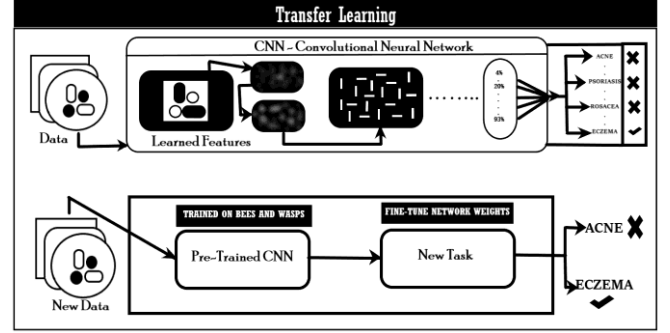


Fig. 10. Transfer learning

The general definition for Domain, Task to understand Transfer Learning definition is as follows, Domain: Let X be the domain and can be defined as $X=\{K, P(L)\}$, where E is feature space, $P(L)$ is marginal distribution and $L = \{l|l_j \in K, j = 1, \dots, n\}$. Task Y is made up of label space W and a decision function g which can be defined as $Y = \{W, g\}$ and g is learned from the data.[35]

The definition of transfer learning is as follows, assuming an observation pertaining to $a^M \in Q^+$ source domain and task i.e., $\{(X_{M_u}, Y_{M_u}) | u = 1, \dots, a^M\}$, and observation about $a^Y \in Q^+$ target domain and task i.e., $\{(X_{N_v}, Y_{N_v}) | v = 1, \dots, a^N\}$, transfer leverages information from the source domain to maximize the efficiency of trained decision functions g^{Y_v} ($v = 1, \dots, a^N$) on the target domain [35][36], where a is number domains, X_M is the source domain, Y_M is the source task, X_N target domain, Y_N is the target task and Q is the Iteration or kernel number.

With the help of pooling and applying convolutions, we can adjust the picture's resolution to our project's requirements. The Inception is very flexible in doing so.

SCREENSHOTS

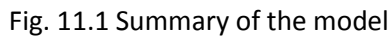
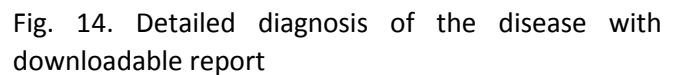
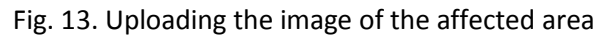


Fig. 12. Entering details of the victim



This paper explained the method used to detect integumentary diseases by employing Deep learning technology. With the help of transfer learning technique and utilizing Inception neural network, high accuracy can be achieved with a minimal number of images in a dataset. This system cannot replace experts, but it can assist the patient in making a preliminary diagnosis. The model summary is as follows, total parameters: 74,513,668 trainable parameters: 52,710,884 non-trainable parameters: 21,802,784. The prototype created using this method predicted integumentary system disorders with 90.28% accuracy.

128

There is ton of scope in research topic. To further improve the model's reliability, regularly spaced shifting and adversarial networks can be employed. The training dataset used in this paper is around one thousand two hundred, to achieve the accuracy above 95% at least for each disease 1500 images i.e., around 7000 images should be used.

References

- [1] O. Trabelsi, L. Tlig, M. Sayadi and F. Fnaiech, "Skin disease analysis and tracking based on image segmentation," 2013 International Conference on Electrical Engineering and Software Applications, 2013, pp. 1-7
- [2] A. Ajith, V. Goel, P. Vazirani and M. M. Roja, "Digital dermatology: Skin disease detection model using image processing," 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), 2017, pp. 168-173
- [3] Wirdayanti et al., "Face Skin Disease Detection with Textural Feature Extraction," 2020 6th International Conference on Science in Information Technology (ICSITech), 2020, pp. 133-137
- [4] Z. Ma and J. M. R. S. Tavares, "A Novel Approach to Segment Skin Lesions in Dermoscopic Images Based on a Deformable Model," in *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 2, pp. 615-623, March 2016
- [5] R. Suganya, "An automated computer aided diagnosis of skin lesions detection and classification for dermoscopy images," 2016 International Conference on Recent Trends in Information Technology (ICRTIT), 2016, pp. 1-5
- [6] K. Roy, S. S. Chaudhuri, S. Ghosh, S. K. Dutta, P. Chakraborty and R. Sarkar, "Skin Disease detection based on different Segmentation Techniques," 2019 International Conference on Opto-Electronics and Applied Optics (Optronix), 2019, pp. 1-5
- [7] J. Rathod, V. Waghmode, A. Sodha and P. Bhavathankar, "Diagnosis of skin diseases using Convolutional Neural Networks," 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2018, pp. 1048-1051
- [8] Singh, S. K., & Jalal, A. S. (2016, August). A robust approach for automatic skin cancer disease classification. In *2016 1st India International Conference on Information Processing (IICIP)* (pp.1-4). IEEE.
- [9] Z. E. Diame, M. N. Al-Berry, M. A. .-M. Salem and M. Roushdy, "Deep Learning Architectures For Aided Melanoma Skin Disease Recognition: A Review," 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), 2021, pp. 324-329
- [10] X. He et al., "Computer-Aided Clinical Skin Disease Diagnosis Using CNN and Object Detection Models," 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 4839-4844
- [11] Christophers, E. (2001). Psoriasis– epidemiology and clinical spectrum. *Clinical and experimental dermatology*, 26(4), 314-320.
- [12] Bahamdan, K., Mahfouz, A. A., Tallab, T., Badawi, I. A., & AL-AMARI, O. M. (1996). Skin diseases among adolescent boys in Abha, Saudi Arabia. *International journal of dermatology*, 35(6), 405-407
- [13] Karski, T. (2018). Biomechanical Aetiology of the So-called Adolescent Idiopathic Scoliosis (AIS). *Lublin Classification (1995-2007)*. *Causative Influences Connected with "Gait" and "Standing 'at ease'on the Right Leg"*. *J Orthop Bone Res 1: 102 Abstract Keywords: So-Called Idiopathic Scoliosis*.
- [14] Joseph, N., Kumar, G. S., & Nelliyanil, M. (2014). Skin diseases and conditions among students of a medical college in southern India. *Indian dermatology online journal*, 5(1), 19.
- [15] Kolkur, S., & Kalbande, D. R. (2016, November). Survey of texture based feature extraction for skin disease detection. In *2016 International Conference on ICT in Business Industry & Government (ICTBIG)* (pp. 1-6). IEEE
- [16] Chakraborty, S., Mali, K., Chatterjee, S., Anand, S., Basu, A., Banerjee, S., ... & Bhattacharya, A. (2017, October). Image based skin disease detection using hybrid neural network coupled bag-of-features. In *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)* (pp. 242-246). IEEE.
- [17] Kolkur, S., Kalbande, D. R., & Kharkar, V. (2018). Machine learning approaches to multi-class human skin disease detection. *International Journal of Computational Intelligence Research*, 14(1), 1-12.
- [18] Balaji, V. R., Suganthi, S. T., Rajadevi, R., Kumar, V. K., Balaji, B. S., & Pandiyan, S. (2020). Skin disease detection and segmentation using dynamic graph cut algorithm and classification through Naive Bayes classifier. *Measurement*, 163, 107922.
- [19] Chakraborty, S., Mali, K., Chatterjee, S., Banerjee, S., Mazumdar, K. G., Debnath, M., ... & Roy, K. (2017, August). Detection of skin disease using metaheuristic supported artificial neural networks. In *2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON)* (pp. 224-229). IEEE.
- [20] Roy, K., Chaudhuri, S. S., Ghosh, S., Dutta, S. K., Chakraborty, P., & Sarkar, R. (2019, March). Skin Disease detection based on different Segmentation Techniques. In *2019 International Conference on Opto-Electronics and Applied Optics (Optronix)* (pp. 1-5). IEEE.
- [21] Lu, Y., & Xu, P. (2018). Anomaly detection for skin disease images using variational autoencoder. *arXiv preprint arXiv:1807.01349*.
- [22] Okuboyejo, D. A., Olugbara, O. O., & Odunaike, S. A. (2013, October). Automating skin disease diagnosis using image classification. In *proceedings of the world congress on engineering and computer science* (Vol. 2, pp. 850-854).
- [23] Ahmad, B., Usama, M., Huang, C. M., Hwang, K., Hossain, M. S., & Muhammad, G. (2020). Discriminative feature learning for skin disease classification using deep convolutional neural network. *IEEE Access*, 8, 39025-39033.
- [24] Gu, Y., Ge, Z., Bonnington, C. P., & Zhou, J. (2019). Progressive transfer learning and adversarial domain adaptation for cross-domain skin disease classification. *IEEE journal of biomedical and health informatics*, 24(5), 1379-1393.
- [25] Liao, H., Li, Y., & Luo, J. (2016, December). Skin disease classification versus skin lesion characterization: Achieving robust diagnosis using multi-label deep neural networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)* (pp. 355-360). IEEE.
- [26] Hegde, P. R., Shenoy, M. M., & Shekar, B. H. (2018, September). Comparison of machine learning algorithms for skin disease classification using color and texture features. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1825-1828). IEEE.

- [27] Hegde, P. R., Shenoy, M. M., & Shekar, B. H. (2018, September). Comparison of machine learning algorithms for skin disease classification using color and texture features. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1825-1828). IEEE.
- [28] Wei, L. S., Gan, Q., & Ji, T. (2018). Skin disease recognition method based on image color and texture features. *Computational and mathematical methods in medicine*, 2018.
- [29] Li, W., Zhuang, J., Wang, R., Zhang, J., & Zheng, W. S. (2020, April). Fusing metadata and dermoscopy images for skin disease diagnosis. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)* (pp. 1996-2000). IEEE.
- [30] Lopez, A. R., Giro-i-Nieto, X., Burdick, J., & Marques, O. (2017, February). Skin lesion classification from dermoscopic images using deep learning techniques. In *2017 13th IASTED international conference on biomedical engineering (BioMed)* (pp. 49-54). IEEE.
- [31] Thurnhofer-Hemsi, K., López-Rubio, E., Domínguez, E., & Elizondo, D. A. (2021). Skin lesion classification by ensembles of deep convolutional networks and regularly spaced shifting. *IEEE Access*, 9, 112193-112205.
- [32] F. P. dos Santos and M. A. Ponti, "Robust feature spaces from pre-trained deep network layers for skin lesion classification," in Proc. 31st SIBGRAPI Conf. Graph., Patterns Images (SIBGRAPI), Oct. 2018, pp. 189–196.
- [33] A. H. Shahin, A. Kamal, and M. A. Elattar, "Deep ensemble learning for skin lesion classification from dermoscopic images," in Proc. 9th Cairo Int. Biomed. Eng. Conf. (CIBEC), Dec. 2018, pp. 150–153.
- [34] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in Proc. NIPS, 2014, pp. 3320–3328.
- [35] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., ... & He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1), 43-76.
- [36] S. Jialin Pan and Q. Yang, "A survey on transfer learning," IEEE Trans. Knowl. Data Eng., vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
- [37] <https://www.kaggle.com/shubhamgoel27/dermnet>