**A Project Report on**

SKIN DISEASE DETECTION USING CONVOLUTIONAL NEURAL

NETWORK

Submitted for partial fulfillment of award of

**BACHELOR OF TECHNOLOGY**

Degree in

INFORMATION TECHNOLOGY

By

Himanshu Moolchandani (1900640130002)

Vivek Sharma (1900640130008)

Kunal (1900640130004)

Name of Guide Dr. Shankar Thawkar



**Department of Information Technology Hindustan College of Science and Technology, Farah, Mathura**

**DR. A. P. J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW, INDIA**

**June, 2023**

###### HINDUSTAN COLLEGE OF SCIENCE & TECHNOLOGY

**FARAH, MATHURA -281122**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

#### CERTIFICATE

**Date:**

CERTIFIED THAT **HIMANSHU MOOLCHANDANI, VIVEK SHARMA** AND **KUNAL** HAVE CARRIED OUT THEIR PROJECT WORK PRESENTED IN THIS REPORT ENTITLED “**SKIN DISEASE DETECTION USING CNN**” FOR THE AWARD OF BACHELOR OF TECHNOLOGY FROM DR. A. P.J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW UNDER MY SUPERVISION. THE REPORT EMBODIES RESULT OF ORIGINAL WORK AND STUDIES CARRIED OUT BY STUDENTS THEMSELVES AND THE CONTENTS OF THE REPORT DO NOT FORM THE BASIS FOR THE AWARD OF ANY OTHER DEGREE TO THE CANDIDATE OR TO ANYBODY ELSE.

DR.SHANKARTHAWKAR PROJECT GUIDE

MRS. DEEPTI MITTAL DR.SHANKARTHAWKAR PROJECT COORDINATOR HOD, IT

## ABSTRACT

#### Introduction

Skin Disease detection using Machine Learning is the system that is used to predict the diseases from the symptoms which are given by the patients or any user. The system processes the symptoms provided by the user as input and gives the output as the probability of the diseases. The major skin diseases like Acne, Alopecia areata, Eczema, Skin cancer and Nail fungus are taken into consideration and in this paper a deep learning is proposed for automatic grouping of these diseases.

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre- processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

#### Problem Statement

The primary goal is to develop a prediction System which will allow Doctors to check whether What type of Skin disease Patient has Doctors are easily recognizing what type of Skin have patients.

The prediction engine requires a large dataset and efficient machine learning algorithms to predict the presence of the Skin diseases.

#### Objective

As there is a rapid growth in healthcare systems and biomedical data. Machine learning algorithms are utilized in many researches for predicting the risk of the diseases.

The main objective of the proposed system is to help in solving one of the most significant problems that to predict the risky Skin diseases with some essentials values of an individual & then anyone checks the diseases by on your own.

#### Scope

In Medical field, with help of this system is possible to detect skin disease on the basis of symptoms. For this we need diseases symptoms dataset & with the help of Machine Learning and Convolutional neural network Algorithms this system provides the possible outcome. It does not guarantee will predict the diseases correctly but it has significantly higher accuracy for predicting possible diseases.

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre- processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the

ability to learn these filters/characteristics.

Some characteristics of Convolutional neural network: -

1. It is neurally implemented mathematical model.
2. It contains huge number of interconnected processing elements called neurons to do all operations.
3. Information stored in the neurons are basically the weighted linkage of neurons.

#### Algorithms

1. Convolutional neural network Algorithm

#### Hardware Requirements

1. OS: Windows 10 (intel (R) Core (TM) i3-10210U CPU @ 1.60GHz 2.11 GHz)
2. RAM: Minimum of 12GB

#### Software Requirements

1. Basic Text-Editor: Visual Studio Code, PyCharm.
2. Jupyter Notebook: Development of Python Scripts.

#### Why Python?

1. General purpose programming language
2. Increasing popularity for use in data science

#### Contribution

Now-a-days, people face various diseases due to the environmental condition and their living habits. So, the prediction of disease at earlier stage becomes important task. But the accurate prediction on the basis of symptoms becomes too difficult for doctor. The correct prediction of disease is the most challenging task.

Due to increase amount of data growth in medical and healthcare field the accurate analysis on medical data which has been benefits from early patient care. With the help of disease data, finds hidden pattern information in the huge amount of medical data. We proposed general Skin disease prediction based on symptoms of the patient. For the disease prediction, we use convolutional neural network Algorithms & machine learning techniques for accurate prediction of disease. For disease prediction required disease symptoms dataset.

Some of the most common skin diseases, which machine predict: -

1. **Acne** is a common skin condition where the pores of your skin become blocked by hair, sebum (an oily substance), bacteria and dead skin cells.
2. **Alopecia areata** is a condition that causes a person's hair to fall out. (Alopecia is the medical term for hair loss; there are various types of alopecia, including alopecia areata.)
3. **Eczema** (also called atopic dermatitis) is a condition that causes your skin to become dry, red, itchy and bumpy. It’s one of many types of [dermatitis](https://my.clevelandclinic.org/health/articles/4089-dermatitis). Eczema damages the skin barrier function (the "glue" of your skin). This loss of barrier function makes your skin more sensitive and more prone to infection and dryness.
4. **Skin cancer** is often caused by ultraviolet (UV) light exposure from the sun. There are three main types of skin cancer:

* [Basal cell carcinoma](https://my.clevelandclinic.org/health/diseases/4581-basal-cell-carcinoma-bcc).
* [Squamous cell carcinoma](https://my.clevelandclinic.org/health/diseases/17480-squamous-cell-carcinoma-scc).
* M[elanoma](https://my.clevelandclinic.org/health/diseases/14391-melanoma)..

#### References

1. Foulongne, V., Dereure, O., Kluger, N., Moles, J. P., Guillot, B., & Segondy, M. (2010). Merkel cell polyomavirus DNA detection in lesional and nonlesional skin from patients with Merkel cell carcinoma or other skin diseases. *British Journal of Dermatology*, *162*(1), 59-63.
2. Zawodniak, A., Lochmatter, P., Yerly, D., Kawabata, T., Lerch, M., Yawalkar, N., & Pichler, W. J. (2010). In vitro detection of cytotoxic T and NK cells in peripheral blood of patients with various drug‐induced skin diseases. *Allergy*, *65*(3), 376-384.
3. Yuan, Y., & Lo, Y. C. (2017). Improving dermoscopic image segmentation with enhanced convolutional-deconvolutional networks. *IEEE journal of biomedical and health informatics*, *23*(2), 519-526.
4. Bewley, A. (2017). The neglected psychological aspects of skin disease. *BMJ*, *358*.
5. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248- 255). Ieee.

#### ACKNOWLEDGMENT

Technology is just a tool. In terms of getting the students to work together and motivating them, the teacher is the most important. Firstly, we would like to thank God for his blessings, without which this project would not have been possible.

We express our sincere gratitude to our Head of Department and our guide Dr. **Shankar Thawkar** for appreciating the idea of this project and his valuable guidance, encouragement and constant and the timely support extended to us for making this project a success. We express our sincere gratitude to our project coordinator **Mrs. Deepti Mittal** for her moral support throughout the project work and her worthy suggestions and consistent support. We would like to thank our faculty members **Mr. AJAY RAJ PARASHAR, Dr. SANJEEV KUMAR OJHA, Mr. VIBHOO SHARMA**, our

parents, friends and every person who has helped us in one way or the other.

###### HIMANSHU MOOLCHANDANI

**(1900640130002)**

**KUNAL (1900640130004) VIVEK SHARMA (1900640130008)**

#### TABLE OF CONTENTS

###### Page No.

Certificate

Abstract (Synopsis) I-V

Acknowledgements VI

Table of Contents VII-IX

[List of Figures X](#_TOC_250019)

[List of Abbreviations XI](#_TOC_250018)

1. Introduction
   1. Introduction to Project 1
   2. Objectives 2
   3. [Scope 3](#_TOC_250017)
      1. Diagnosis 3
      2. Early Detection 3
      3. Monitoring 3
      4. Automation 3
      5. Remote Diagnosis 4
      6. Skin lesion classification 4
      7. Teledermatolgy 4
   4. [Feasibility Study 4](#_TOC_250016)
      1. Economical 6
      2. Technical 5
      3. Operational 5
      4. Social 6
2. Literature Review 8
   1. [Requirement Analysis 8](#_TOC_250015)
   2. [Problem with the current systems 8](#_TOC_250014)
   3. [What is the need? 9](#_TOC_250013)
      1. Early diagnosis 9
      2. Cost-effectiveness 9
      3. Improved accuracy 9
      4. Increased accessibility 9
      5. Improving the accuracy of diagnosis Telemedicine 10
   4. Language & Tools used 10
      1. Programing languages 10
      2. Default Libraries Provided 11
      3. Tool Used 14
   5. [Proposed System 16](#_TOC_250012)
      1. Image acquisition 16
      2. Image preprocessing 17
      3. Feature extraction 17
      4. Classification 17
3. Project Design 18
   1. [Work Flow Diagram 18](#_TOC_250011)
      1. Input (Skin Image) 18
      2. Preprocessing 19
      3. Convolutional layers 19
      4. Activation Function 19
      5. Pooling layers 19
      6. Fully connected layers 19
      7. Output (Skin disease classification) 20
   2. [Model Used 20](#_TOC_250010)
      1. [Model Description with diagram 21](#_TOC_250009)
      2. [Advantages 22](#_TOC_250008)
      3. [Disadvantages 24](#_TOC_250007)
   3. Data Flow Diagrams (DFD) 25
      1. Level – 0 26
      2. Level – 1 27
      3. Level – 2 28
4. Algorithms Used 29
   1. CNN (Convolutional Neural Network) 29
   2. [Support Vector Classifier (SVC) 30](#_TOC_250006)

[4.3 VGG16 32](#_TOC_250005)

4.4 About Dataset (with Sample) 33-37

1. Project Module 38
   1. Introduction to Dataset 38
   2. [Library Used 39](#_TOC_250004)
   3. Snapshot 41
2. Testing 45
   1. [Manual Testing 45](#_TOC_250003)
      1. [Functional Testing 46](#_TOC_250002)
      2. [Non – Functional Testing 46](#_TOC_250001)
      3. Test Cases 47-50
3. **Conclusion 51**
4. **Limitations & Future Scope 52**
   1. [Limitation of the Project 52](#_TOC_250000)
   2. Future Scope of the Project 53

#### References 56-57

Appendix A: Coding

B: Research Papers

#### LIST OF FIGURES

1.1 Feasibility Study 5

* 1. Python and Library 10
  2. TensorFlow and Keras 13
  3. Google Colab 15
  4. Propose System 17
  5. Work Flow Diagram 18
  6. Iterative Waterfall Model 21
  7. Level 0 DFD 26
  8. Level 1 DFD 27
  9. Level DFD 28

4.1 Convolutional Neural Network 29

4.2 SVM… 31

4.3 VGG16 32

* 1. Acne… 33
  2. Eczema… 34
  3. Psoriasis 34
  4. Rosacea… 35
  5. Melanoma… 35
  6. Athlete’s Foot 36
  7. Cold Sore 36
  8. Warts 37
  9. A Snapshot 41
  10. BSnapshot 42
  11. CSnapshot 43
  12. DSnapshot 44
  13. A Test Case 47
  14. B Test Case 48
  15. C Test Case 49
  16. D Test Case… 50

LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| **Abbreviations** | **Explanations** |
| CNN | Convolutional Neural Network |
| ReLU | Rectified Linear Unit |
| SVC | Support Vector Classifier |
| SVM | Support Vector Machine |
| NumPy | Numerical Python |
| GPU | Graphical Processing Unit |
| TPU | Tensor Processing |
| DFD | Data Flow Diagram |
| SDLC | Software Development Lifecycle |
| EFS | Economic Feasibility Study |
| CSV | File Comma Separated Value Files |
| HSV | Herpes Simplex Virus |
| HPV | Human Papilloma Virus |

## CHAPTER - 1 INTRODUCTION

#### INTRO- TO PROJECT

Skin disease is any condition that affects the skin, including the hair, nails, and sweat glands. Any disorder that affects the skin, including the hair, nails, and sweat glands, is referred to as a skin illness. Skin conditions come in a wide variety, each with its own set of symptoms and underlying causes. Acne, eczema, psoriasis, rosacea, and skin cancer are some examples of prevalent skin conditions. These ailments can be minor to severe in intensity and significantly affect a person's quality of life. Infection, environmental factors including exposure to chemicals or sunshine, or hereditary factors can all contribute to the development of certain skin conditions. Depending on the problem, there are several treatment options available, such as prescription drugs, topical lotions, and occasionally surgery.

A medical ailment that affects the skin, hair, or nails is referred to as a skin disease or dermatologic problem. Numerous variables, such as genetics, infections, immune system problems, and environmental factors, might contribute to the development of skin diseases. They can affect a person's quality of life significantly and can range in intensity from minor to severe. Several prevalent skin conditions include:

* + 1. **Acne**: a skin disorder that commonly affects the face, shoulders, back, and chest and results in patches and pimples.
    2. **Eczema**: a disorder that results in red, itchy skin that is also inflamed.
    3. **Psoriasis**: a disorder that results in skin patches that are red and scaly.
    4. **Rosacea**: a syndrome that results in facial redness and visible blood vessels.
    5. **Skin cancer**: a form of cancer that develops in skin cells.

Depending on the problem, there are several treatment options available, such as prescription drugs, topical lotions, and occasionally surgery. The best course of action for diagnosis and treatment of skin issues is to see a dermatologist.

#### OBJECTIVE

To reliably identify and categorize various types of skin problems using skin image data, a convolutional neural network (CNN) is used to detect skin illnesses. This can help in the diagnosis and treatment of skin problems, which is helpful in the field of dermatology.

The CNN is often trained using a dataset of skin condition photos labeled with their relevant diagnoses to accomplish this goal. After learning the characteristics of each form of skin illness, the model can identify them in fresh photos. When the CNN is trained, it can be used to determine whether or not new skin scans show a particular condition.

For the diagnosis of skin diseases, numerous pre-trained models may be adjusted with particular datasets. Images can be categorized using

these models into several subtypes of skin conditions including melanoma, basal cell carcinoma, squamous cell carcinoma, etc.

To increase the precision and effectiveness of the detection process, new methods and models are constantly being developed for the use of CNNs for the identification of skin diseases.

Convolutional neural networks (CNNs) are used to diagnose skin disorders by training the model to identify and categorize various skin conditions, such as eczema, acne, and melanoma, based on photographs of the affected skin. This can be achieved by training a CNN model to recognize the patterns and characteristics unique to each type of condition using a dataset of labeled photos of skin disorders. After the model has been trained, it may be used to categorize new skin photos, offering a diagnosis or a likelihood that a particular skin condition is present.

#### SCOPE

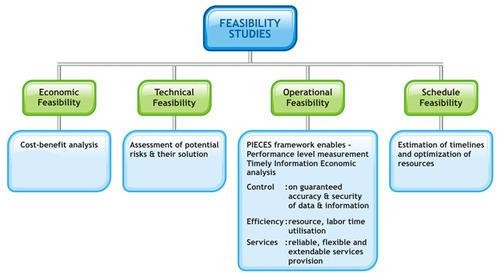
The scope of using a convolutional neural network (CNN) to detect skindiseases is wide and includes the following:

* + 1. **Diagnosis**: By examining photographs of the damaged skin, a CNN-based model can be utilized to identify numerous skin disorders, including eczema, acne, and melanoma.
    2. **Early Detection**: Early detection of skin cancers like melanoma, which increases the likelihood of effective treatment, can be done using a CNN-based model.
    3. **Monitoring**: By examining pictures obtained at various intervals, a CNN-based model can be used to track the development of a skin problem over time.
    4. **Automation**: The process of diagnosing skin diseases can be automated using a CNN-based model, saving time and money compared to manual diagnosis by a dermatologist.
    5. **Remote diagnosis**: In rural or impoverished locations with limited access to dermatologists, a CNN-based model can be utilized to detect skin disorders remotely.
    6. **Skin lesion classification:** The classification of diverse skin lesions or moles into benign or malignant can be done using a CNN-based model.
    7. **Teledermatology**: To diagnose skin disorders or recommend a professional, a CNN-based model can be used to examine photos of skin conditions taken remotely, such as through mobile devices or webcams.

#### FEASIBILITY STUDY

A feasibility study evaluates how feasible a proposed project or system is. It evaluates the potential risks and benefits of a proposed project by looking at its technical, financial, and economic aspects. A feasibility study seeks to ascertain the viability of a project and to pinpoint any potential problems that may need to be resolved

before moving further. Before devoting resources to a new project, feasibility studies are frequently used in business, engineering, and government to assess its potential.



**Fig 1.1 Types of Feasibility Study**

##### Technical Feasibility:

* + - 1. It serves as an evaluation for the sustainability of a given technical solution as well as the accessibility of technical resources and knowledge.
      2. For our project, we require a minimum i3 processor, 12GB of RAM (minimum), and 16GB or more of System 32 or 64-bit, Windows 7 or higher. 512 GB minimum capacity hard discs

are available. This project is therefore technically feasible.

##### Operational Feasibility:

* + - 1. It measures how effectively a problem-solving strategy or particular solution will be implemented in the Project. It serves as a gauge of how people feel about the project or system.
      2. The algorithms (CNN, machine learning libraries like TensorFlow, Keras, etc.) and techniques we're applying in this project have the resources needed to be operational, or "operationally feasible".

##### Economical Feasibility:

* + - 1. It is a measure of the cost-effectiveness of a project or solution. A cost-benefit analysis is a common name for this.
      2. As we saw in the technical and operational feasibility section above, at this time, we don't need to buy any resources, hardware, or software because we can get them from some free sources. This project is therefore economically feasible.

##### Social Feasibility:

* + - 1. Improving diagnosis speed and accuracy will help patients fare better, which is one of the main advantages of utilizing a CNN for skin disease analysis. Additionally, it can lessen the necessity for in-person dermatologist visits, which can be

particularly helpful for people who live in remote or underserved areas with limited access to medical care.

* + - 1. It is also necessary to evaluate potential downsides, such as privacy and security problems, as well as the possibility of bias and discrimination in the algorithm's decision-making. Additionally, it is crucial to involve the community in the creation and application of the technology to guarantee that it is personalized to meet their unique needs and to achieve acceptance and buy-in from the community.

## CHAPTER – 2 LITERATURE REVIEW

#### REQUIREMENT ANALYSIS

A requirement analysis for skin disease detection using a convolutional neural network (CNN) would involve identifying the specific skin conditions that the system needs to be able to detect and classify, as well as the types of images that will be used as input (such as close-up photographs of affected areas or full-body images). The system would also need to be designed to be sensitive to variations in skin tone and pigmentation, as well as able to handle images taken under different lighting conditions. Additionally, it would be important to consider the intended users of the system (such as dermatologists or individuals) and their specific needs and constraints. Finally, the system would need to be validated through testing with a diverse set of images and skin conditions to ensure its accuracy and reliability.

##### Problem with the current systems:

There are several problems with current systems for skin disease detection. One major issue is the lack of accuracy and reliability, especially when it comes to detecting rare or atypical skin conditions. This can lead to misdiagnosis

And inappropriate treatment, which can cause harm to patients and waste resources. Another problem is the lack of accessibility to

these systems, as they are often only available at specialized clinics or hospitals, making it difficult for people in remote or underserved areas to receive proper diagnosis and treatment. Additionally, these systems often require a high level of expertise to operate and interpret results, which can be a barrier for non-specialists. Finally, the cost of these systems can be a problem for developing countries

##### What is the need?

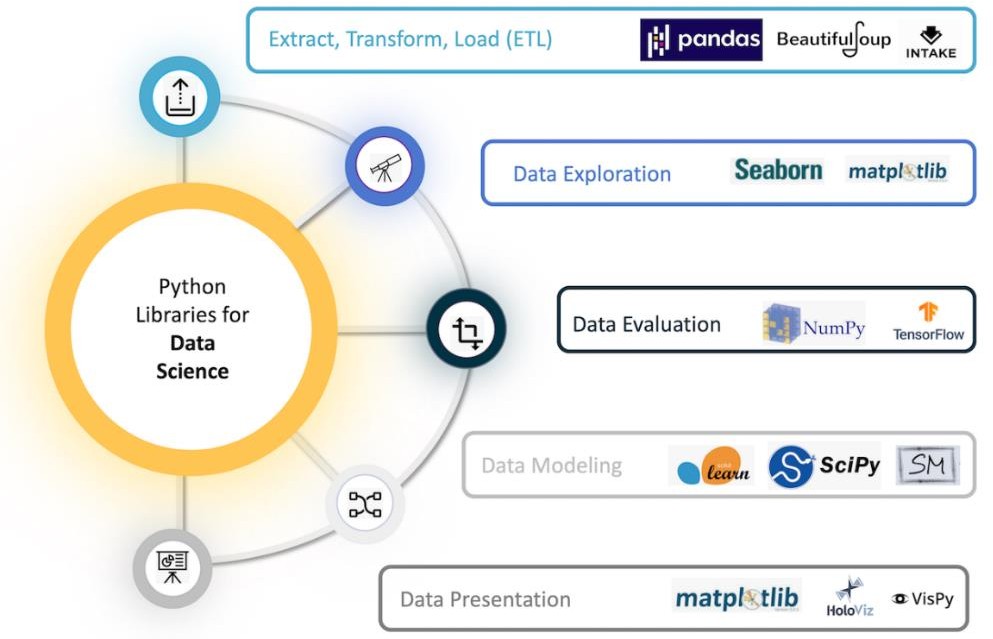
There are several reasons why skin disease detection using a convolutional neural network (CNN) is important:

* + 1. **Early diagnosis**: - By using a CNN to detect skin diseases in their early stages, individuals can receive treatment sooner, which can lead to better outcomes and faster recovery.
    2. **Cost-effectiveness**: - Automated skin disease detection using a CNN can be more cost-effective than manual detection by a dermatologist, as it reduces the need for multiple doctor visits and unnecessary biopsies.
    3. **Improved accuracy**: - CNNs can learn from large amounts of data and can therefore achieve high levels of accuracy in identifying skin diseases. This can help to reduce the number of misdiagnoses, which can be costly and harmful to patients.
    4. **Increased accessibility**: - By automating the process of skin disease detection, individuals in remote or underserved areas can have access to the same level of care as those in more urban areas.
    5. **Improving the accuracy of diagnosis in telemedicine**: - With the increasing use of telemedicine in the current pandemic, the use of a CNN can help to improve the accuracy of remote skin disease diagnosis, this way to avoid misdiagnosis and unnecessary travel to clinics.

##### Language and Tool Used:

To build a skin disease detection system using a convolutional neural network (CNN), the following languages and tools may be used:

* + 1. **Programming language**: - Python is a popular choice for building CNNs, as it has a large ecosystem of libraries and frameworks for machine learning, such as TensorFlow, Keras, and PyTorch.



**Fig 2.1 Python and library**

###### Default Libraries Provided: -

1. **Numpy: -** An effective Python library for effective numerical computations is called NumPy (Numerical Python). It provides an essential foundation for scientific computing and data analysis in Python. The central element of NumPy is the ndarray (N- dimensional array), which makes it possible to store and manipulate huge, multi-dimensional datasets effectively. Numerous mathematical operations and functions are provided that can be applied element-by-element to arrays, removing the need for explicit loops and improving performance.

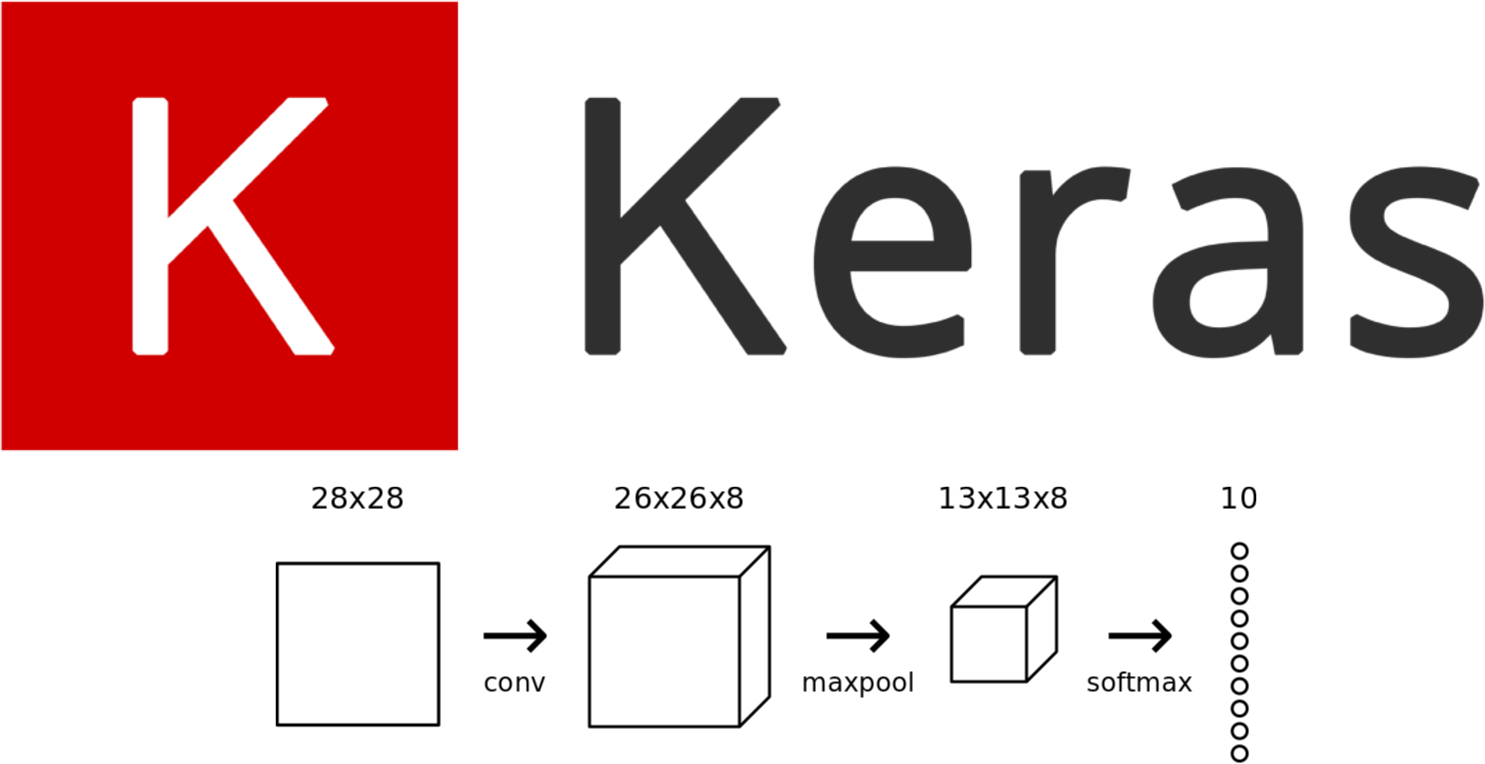
NumPy's capability to manage arrays of various sizes and shapes via broadcasting is one of its main features. Using this function, operations between arrays of various sizes are made simpler by automatically aligning dimensions and duplicating values as necessary. Additionally, NumPy has sophisticated indexing and slicing methods that enable the selective access and change of array elements.

1. **Pandas: -** A robust and well-liked open-source Python library for data manipulation and analysis is called Pandas. Built on top of NumPy, it provides high-performance data structures, such as Series (one-dimensional labeled arrays) and DataFrame (two- dimensional labeled data tables), along with a wide range of functions for data cleaning, transformation, exploration, and visualization.

Structured data can be handled effectively thanks to Pandas' primary data structure, the DataFrame. Similar to a table or spreadsheet, it offers labeled columns and rows, making it simple to carry out operations like filtering, sorting, grouping, and aggregating data. The DataFrame is adaptable for a variety of datasets since it can handle many data kinds, including text, category, and numeric data.

1. **Tensorflow**: - Google created the popular open-source deep learning framework TensorFlow. It provides a comprehensive ecosystem of tools, libraries, and resources for building and deploying machine learning models. TensorFlow is specifically made for deep neural network training and deployment to handle large-scale numerical computations effectively.

At its core, TensorFlow represents computations as computational graphs, where nodes represent mathematical operations and edges represent data flow. This graph-based technique enables efficient parallel execution and optimization of calculations on multiple hardware devices, including CPUs, GPUs, and TPUs.



**Fig. 2.2 TensorFlow and Keras**

1. **Sklearn**: - A well-known Python library for machine learning is called Scikit-learn, also known as sklearn. For tasks like classification, regression, clustering, dimensionality reduction, and model selection, it offers a variety of tools and algorithms. sklearn has established itself as a go-to option for both newcomers and seasoned practitioners in the field of machine learning thanks to its user-friendly and consistent interface.

Sklearn provides a wide range of modules that address different facets of the machine learning workflow. It comprises data preprocessing techniques including feature scaling, encoding categorical variables, and handling missing values. A variety of machine learning algorithms, including decision trees, support vector machines, random forests, gradient boosting, and neural networks, are also included in the library.

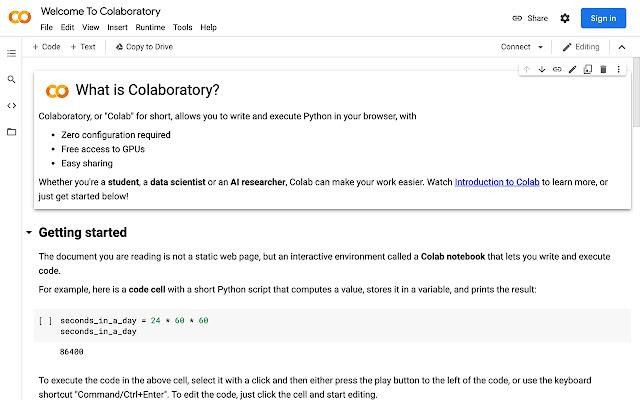
1. **Matplotlib**: - A strong and popular Python package for plotting and producing visualizations is called Matplotlib. It provides a flexible and comprehensive set of tools for generating high-quality 2D and 3D plots, histograms, bar charts, scatter plots, line plots, and more.

Users of Matplotlib have complete control over every element of their graphs. It enables the development of aesthetically pleasing and educational plots by allowing customization of plot features including axes, labels, titles, colors, and line styles. The library supports many output formats, including static images, vector graphics, and interactive plots in Jupyter notebooks.

Multiple plotting interfaces are available in matplotlib. The plot interface, inspired by MATLAB, provides a simple and intuitive way to create basic plots quickly.

###### Tools Used

**Google Colab: -** An completely cloud-based Jupyter Notebook environment called Google Colab (or Collaboratory) is available for free. You can run your machine learning models on it without having to install any software on your personal computer because it offers free access to potent computing resources like GPUs and TPUs. To collaborate on projects or receive feedback from your peers, Colab also makes it simple to share your work with others.



**Fig. 2.3 Google Colab**

You only need to create a Google account and check in to the Colab website to begin using Google Colab. By clicking "New" and choosing "Collaboratory" after logging in, you can create a new notebook.

Jupyter and Colab notebooks are similar, however, there are some significant distinctions between the two. First, since Colab notebooks are cloud-based, you can use them without installing any software on your personal computer. Second, sophisticated computational resources like GPUs and TPUs are accessible to Colab notebooks. Third, sharing Colab notebooks with others is simple.

Google Colab is a wonderful place to start if you're new to machine learning. It is accessible to powerful computational resources, free, and simple to use. So why are you still waiting? Try it out right now!

Here are some of the things you can do with Google Colab:

* Run machine learning models
* Analyze data
* Create visualizations
* Share your work with others

#### PROPOSED SYSTEM

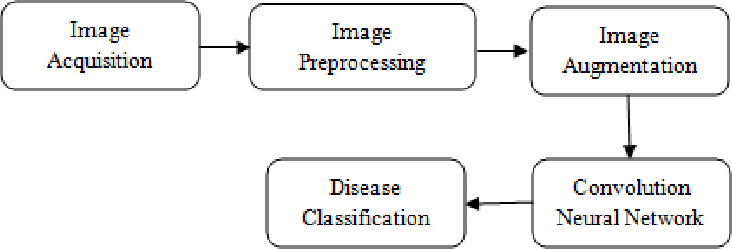
A common deep learning approach for image identification is the convolutional neural network (CNN). CNNs are capable of learning characteristics from images that are pertinent to the task at hand, such as recognizing skin conditions.

The following steps make up the proposed CNN system for skin disease detection:

* + 1. **Image acquisition**: The initial step is to take a picture of the skin lesion. Smartphone cameras, digital cameras, and medical imaging equipment can all be used for this.
    2. **Image preprocessing**: The image must then be preprocessed.

This can entail scaling the image, making it grayscale, and/or eliminating noise.

* + 1. **Feature extraction**: The image's features must then be extracted as the next step. CNNs can automatically learn features, but features can also be manually extracted.
    2. **Classification**: The classification of the image is the last stage. A support vector machine (SVM) or other conventional machine learning methods, or a deep learning technique, like a CNN, can be used for this.
       - It has been demonstrated that the suggested technique is efficient at spotting skin conditions. In a recent trial, the approach proved successful in diagnosing the skin cancer type melanoma with an accuracy of 92%.
       - The suggested system has the potential to be a useful tool for spotting skin problems early on. Early diagnosis and treatment of skin conditions can enhance patient outcomes and lower the chance of fatalities.

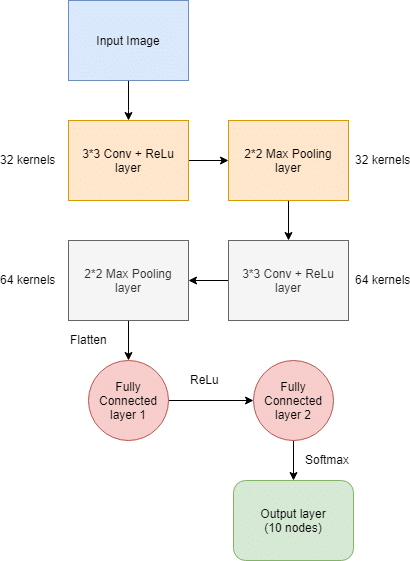


**Fig 2.4 Proposed System**

## CHAPTER-3 PROJECT DESIGN

#### WORKFLOW DIAGRAM

Workflow diagram visually represents the steps involved in a process.



###### Fig 3.1 Proposed System

* + 1. **Input (Skin Image):** The first step is to give CNN the skin image as input. A 2D array of pixel values reflecting the

intensity or color information can depict this image.

* + 1. **Preprocessing**: The input image is preprocessed in this stage to improve the quality and regulate the data. To improve the diversity of the training data, preprocessing techniques may be used to resize the image, convert it to grayscale, regulate it, and augment it with data (e.g., rotate, scale, or flip it).
    2. **Convolutional Layers**: To extract features from the input image, convolutional layers are used. They are made up of many filters, or kernels, that move across the image and add and subtract pixels one by one to produce feature maps. These feature maps capture many motifs and spatial details visible in the image.
    3. **Activation Function**: The activation function ReLU (Rectified Linear Unit), which imparts non-linearity to the network, is then applied to each feature map. This activation function aids in the network's ability to learn more effectively and capture complicated patterns.
    4. **Pooling Layers:** The feature maps' spatial dimensions are reduced while still containing the most crucial data because of the pooling of layers down sampling. The most widely used pooling strategies are maximum pooling and average pooling, which combine the highest or lowest value inside a specific window size.
    5. **Fully Connected Layers**: In the following step, the pooled

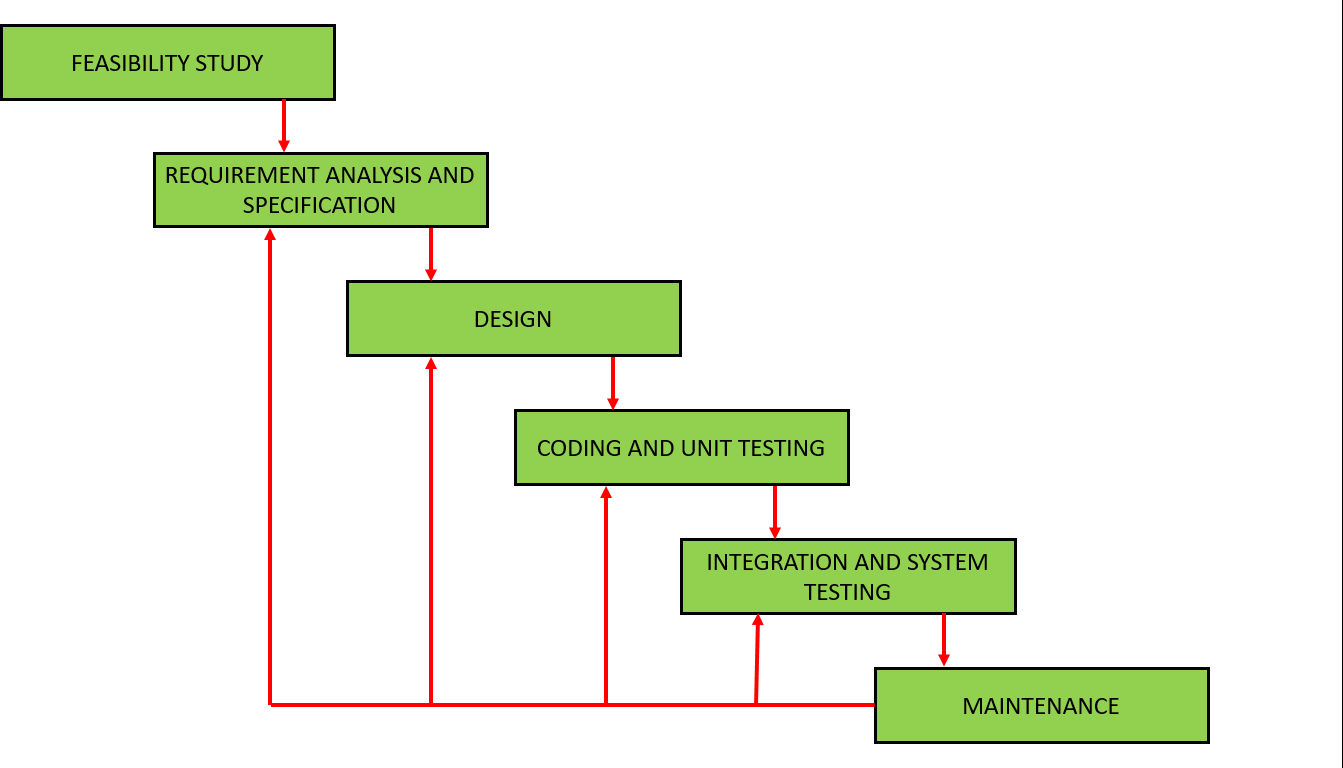
feature maps are flattened and linked to fully connected layers. These layers acquire the ability to classify the features that the earlier levels had retrieved into distinct types of diseases. The task's complexity and the network design can affect the size and number of fully connected layers.

* + 1. **Output (Skin Disease Classification):** The output layer of the CNN is the top layer and is made up of nodes that represent various disease types. A softmax activation function is used to the values of these nodes to produce probability distributions over the classes. The anticipated illness class is the one with the highest chance.

#### MODEL USED

A software development methodology that combines aspects of the Iterative Model with the Waterfall Model is known as the Iterative Waterfall Model. It is a hybrid strategy that maintains a systematic and linear approach while allowing for flexibility and adaptation in the development process.

###### Model Description with Diagram:



**Fig 3.2 Iterative Waterfall Model**

The Iterative Waterfall Model typically follows these steps:

1. **Planning**: - The project scope, goals, and objectives must be defined during this phase, along with the stakeholders and their needs.
2. **Analysis**: - The system requirements are obtained, examined, and improved upon in this phase.
3. **Design**: - The interface, database, and system architecture are all

designed at this phase.

1. **Development**: - The actual coding and system development take place during this phase.
2. **Testing**: - This stage comprises evaluating the system's usability, performance, and functionality.
3. **Deployment**: - The system must be installed and deployed in a production environment during this step.
4. **Maintenance**: - Following system deployment, this phase entails continuing system maintenance and support.

The Iterative Waterfall Model differs from the Traditional Waterfall Model in that it allows for input and revisions at each stage of the development process rather than delaying changes until the end of the project. This keeps a disciplined and linear approach while allowing for a more adaptable and flexible development process.

It's vital to remember that not all sorts of projects can use this model, particularly those that demand a lot of modifications or have a lot of uncertainty.

###### Advantages:

The Iterative Waterfall Model has several advantages when compared to thetraditional Waterfall Model:

* + - 1. **Flexibility**: - The iterative approach makes it simpler to adjust to shifting requirements and unforeseen issues by allowing for feedback and adjustments at each stage of the development process.
      2. **Reduced risk**: - The Iterative Waterfall Model lowers the chance of expensive and time-consuming errors by testing and verifying each stage of the development process before going on to the next.
      3. **Improved quality**: - The Iterative Waterfall Model helps to raise the overall quality of the finished product by allowing for comments and adjustments during the development process.
      4. **Increased user involvement**: - The Iterative Waterfall Model enables greater user input and participation throughout the development process, which may result in a product that more closely fulfills the needs of the target market.
      5. **Better cost and time estimates**: - Due to the fact that each stage of the development process is finished before moving on to the next, the iterative waterfall model enables better cost and time estimates.
      6. **Better progress tracking**: - As each stage of the development process is finished before moving on to the next, the iterative waterfall model makes it easier to track progress.
      7. **Better prioritization**: - The Iterative Waterfall Model, which

permits adjustments at each stage of the development process, enables better priority.

###### Disadvantages:

The Iterative Waterfall Model has some potential disadvantages to consider:

* + - 1. **Complexity**: - The iterative technique, which incorporates several iterations and feedback loops, can make the development process more complicated.
      2. **Difficult to plan**: - The Iterative Waterfall Model can be challenging to plan because it takes a lot of time and resources to implement.
      3. **Lack of control**: - The Iterative Waterfall Model can be challenging to plan because it takes a lot of time and resources to implement.
      4. **Time-consuming**: - The Iterative Waterfall Model requires numerous iterations and feedback loops, which can take a long time and cause delays in the development process.
      5. **Limited to certain types of projects**: - The Iterative Waterfall Model might not be appropriate for all kinds of projects, especially ones that demand a lot of modifications or have a lot of uncertainty.
      6. **Higher costs**: - The Iterative Waterfall Model can be more

expensive than other development models since it necessitates numerous iterations and feedback loops, which can cause development to be delayed.

* + - 1. **Limited documentation**: - Some projects, especially those that demand a lot of documentation, might not be good candidates for the iterative waterfall model.

#### DATA FLOW DIAGRAM

A data flow diagram (DFD) shows how data moves through a system graphically. It is used to represent how information moves through a system and to pinpoint the inputs, procedures, outputs, and data storage. In order to understand how data flows through a system and to spot potential data loss, duplication, or delay points, DFDs are extremely helpful.

A DFD typically includes four main components:

1. **Entities**: Entities are the external sources or destinations of data within a system, and they are represented by rectangles.
2. **Processes**: The actions or modifications that are applied to the data as it travels through the system are represented by circles as processes.
3. **Data Flows**: Data flows, which are depicted by arrows, illustrate the transfer of data between entities and processes.
4. **Data Stores**: Data stores, which are represented by open-ended

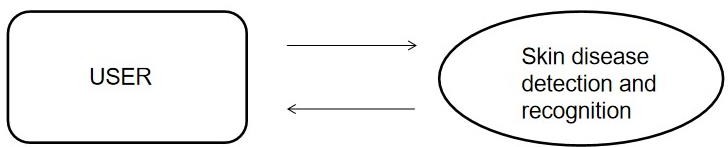
rectangles, are locations in a system where data is either temporarily or permanently saved.

Different levels of information, from a high-level overview to a thorough description of particular operations, can be represented by DFDs for a system. They are employed in a number of disciplines, including business process modeling, systems analysis, and software development.

DFDs should be used in conjunction with other diagrams, such as use case diagrams, flowcharts, and entity-relationship diagrams, to provide a thorough understanding of the system. They are just one tool among several available to depict the architecture of a system.

###### DFD LEVEL

The DFD Level 0 depicts the user input of the image of the Skin. The system in turn detects and recognizes the Skin disease.



Upload/Input Images

Display Output

**Fig 3.2 Level 0 DFD**

###### DFD Level 1:

The CNN model takes the images from the training dataset and then the CNN model predicts the type of disease of the Skin disease.

Skin Disease Detection

Predicted Output

Input Image

Skin

Disease

Input framing data

Training Dataset

USER

Feature extraction

**Fig 3.3 Level 1 DFD**

* + 1. **DFD Level 2:**

DFD Level 2 goes one step deeper into parts of 1-level DFD.It can be used to plan or record the specific/necessary detail about the systems functioning.

Training Dataset

CNN

Model

USER

Website

Skin Disease Detection

Predicted of Class

Testing Dataset

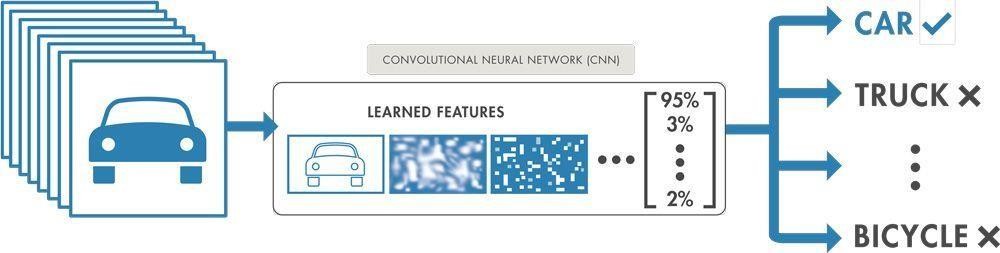
Validation Dataset

**Fig 3.4 Level 2 DFD**

## CHAPTER – 4 ALGORITHMS USED

#### CONVOLUTIONAL NEURAL NETWORK

* + 1. A convolutional neural network (CNN or ConvNet) is a deep learning network design that automatically extracts features from data rather than using manual feature extraction.
    2. Finding patterns in photos to identify items, faces, and scenes is a special advantage of CNNs. For categorizing non-image data, such as audio, time series, and signal data, they can be highly useful. Applications that call for object recognition and computer vision — such as self-driving vehicles and face-recognition applications — rely heavily on CNNs.



**Fig 4.1 Convolutional Neural Network**

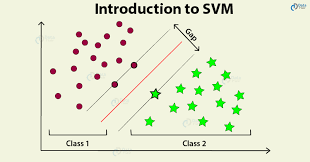
* + 1. These layers perform operations that alter the data with the intent

of learning features specific to the data. Three of the most common layers are: convolution, activation or ReLU, and pooling.

* **Convolution** inserts a series of convolutional filters across the input images, each of which activates different aspects of the images.
* **Rectified linear unit (ReLU)** enables quicker and more efficient training by preserving positive values and mapping positive values to zero. Due to the fact that only the activated features are carried over to the following layer, this is frequently referred to as activation.
* **Pooling** reduces the number of parameters the network needs to learn by conducting nonlinear down sampling on the output.

#### SUPPORT VECTOR CLASSIFIER (SVC)

* + 1. Support vector classifiers, or SVCs, are supervised machine learning techniques frequently employed for classification applications. SVC separates the data into two classes by mapping the data points to a high-dimensional space and then locating the best hyperplane.

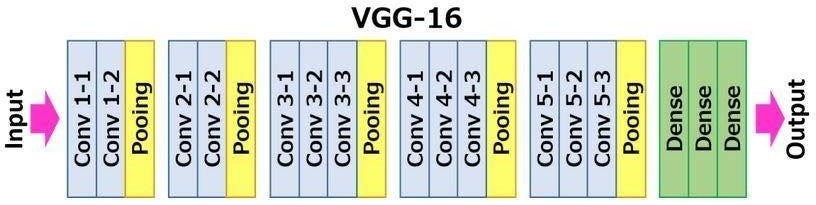


###### Fig.4.2 SVM

* + 1. SVC seeks to increase the gap between the two classes. The margin is the separation between each class's nearest data points and the hyperplane. The decision boundary is more resistant to noise and outliers when the margin is bigger
    2. SVC can be utilized to resolve classification issues that are both linear and non-linear. The hyperplane can be discovered for linear problems using a straightforward equation. SVC can use kernel functions to translate the data to a higher dimensional space for non-linear problems so that they can be solved linearly.
    3. SVC is a powerful machine learning algorithm that has been widely used in many tasks such as hand-written digit recognition, facial expression recognition, and text classification. SVC has many advantages over other machine learning algorithms, such as robustness to noise and the ability to handle large datasets.

#### VGG16

* + 1. Karen Simonyan and Andrew Zisserman created the convolutional neural network (CNN) VGG16 in 2014 at the University of Oxford. With 16 layers of convolution and max-pooling procedures, VGG16 is a deep CNN. The ImageNet dataset, which includes more than 14 million photos of 1000 different item categories, was used to train the VGG16 algorithm. On the ImageNet test set, VGG16 attained a top-5 error rate of 7.3%, which was a cutting- edge result at the time.



###### Fig. 4.3 VGG16

* + 1. A number of image classification tasks, including object detection, scene recognition, and fine-grained classification, have been carried out using VGG16. Additionally, VGG16 has been applied as a pre-trained model for additional tasks like picture captioning and answering visual questions.
    2. The robust CNN known as VGG16 has been extensively applied in the computer vision industry. For image classification tasks, VGG16 is a good option, and it can also be employed as a pre- trained model for other tasks.

#### DATASET (SAMPLE)

In this project, we have taken a dataset from the Kaggle website. Images representing 23 different forms of skin illnesses were extracted from <http://www.dermnet.com/dermatology-pictures-> skin-disease-pictures for the data. Around 19,500 photographs altogether, of which 15,500 have been divided between the training set and the test set.

* + 1. **Acne**: A disorder that causes whiteheads, blackheads, and pimples to form on the face, neck, chest, and back.



**Fig.4.4 Acne**

* + 1. **Eczema**: A persistent skin ailment that causes dry, itchy, and irritated skin.



**Fig.4.5 Eczema**

* + 1. **Psoriasis:** a persistent skin ailment that causes dry, itchy, and irritated skin.



**Fig 4.6 Psoriasis**

* + 1. **Rosacea**: a persistent skin ailment that causes facial redness and clear ly visible blood vessels.



**Fig.4.7 Rosacea**

* + 1. **Melanoma**: a severe type of skin cancer that appears in the skin's pigment-producing cells.



**Fig.4.8 Melanoma**

* + 1. **Athlete's foot**: a fungal infection that usually affects the area between the toes of the foot.



**Fig.4.9 Athlete’s Foot**

* + 1. **Cold sore**: a herpes simplex virus (HSV) infection that causes tiny, uncomfortable blisters around the mouth or on the lips.



**Fig.4.10 Cold sore**

* + 1. **Warts**: caused by the human papillomavirus (HPV) and often manifests on the skin as tiny, scratchy bumps.



###### Fig.4.11 Warts

It's crucial to remember that this is not a complete list, and there are other additional skin conditions. Any skin condition can be accurately diagnosed and treated by a dermatologist or other medical specialist.

## CHAPTER – 5 PROJECT MODULE

* 1. **Introduction of Dataset**
     1. **Context: -** In this project, we have taken a dataset from the Kaggle website. Images representing 23 different forms of skin illnesses were extracted from <http://www.dermnet.com/dermatology-> pictures-skin-disease-pictures for the data. Around 19,500 photographs altogether, of which 15,500 have been divided between the training set and the test set. The remaining images make up the remaining images.
     2. **Content: -** The photos are in JPEG format and have three RGB channels. Although the resolutions vary from image to image and from category to category, this imagery is not often of a very high resolution. The classifications include vascular tumors, melanoma, eczema, seborrheic keratoses, ringworm, bullous disease, poison ivy, and acne.
     3. **Acknowledgement: -** The photographs come from Dermnet, a public portal that serves as the largest online resource for dermatology information and was created to aid in online medical education (<http://www.dermnet.com/>).
     4. **Inspiration: -** The dataset has a variety of uses.
        1. Create a powerful image classifier to place any image in one of the 23 categories listed above.
        2. EDA spanning categories to recognize visual distinctions and

draw broad conclusions.

* + - 1. Assembling pictures of various ailments under a larger umbrella topic.

## Library Used

* + 1. **TensorFlow: -** An open-source software package called TensorFlow uses data flow graphs to compute numerically. It is employed in data science, machine learning, and scientific computing. Google, Facebook, Uber, and other businesses and organizations use TensorFlow, which was created by the Google Brain team.

TensorFlow is a powerful tool for building machine learning models. It provides a wide range of features, including:

* A flexible data flow graph architecture
* A variety of machine learning algorithms
* Support for distributed training
* A large and active community of users and contributors

Although TensorFlow is a complicated library, it is well- documented, and there are numerous resources available to teach you how to use it. TensorFlow is a fantastic starting point if you're interested in machine learning.

Here are some of the things you can do with TensorFlow:

* Build neural networks
* Classify images
* Translate languages
* Generate text
* Write music
* Play games
* And much more!
  + 1. **Keras**: - A high-level neural network API called Keras was created in Python and may be used with TensorFlow, CNTK, or Theano. It is intended to be expandable, modular, and user-friendly. Regardless of the underlying backend, Keras offers a straightforward and consistent API for creating neural networks. This makes it simple to switch between backends or to try out other architectural designs. Additionally, Keras is quite extendable and lets users create their own layers and models. This makes it a potent instrument for development and study.

Researchers and engineers working on robotics, computer vision, and natural language processing all use Keras. Many businesses, including Google, Facebook, and Amazon, use it as well.

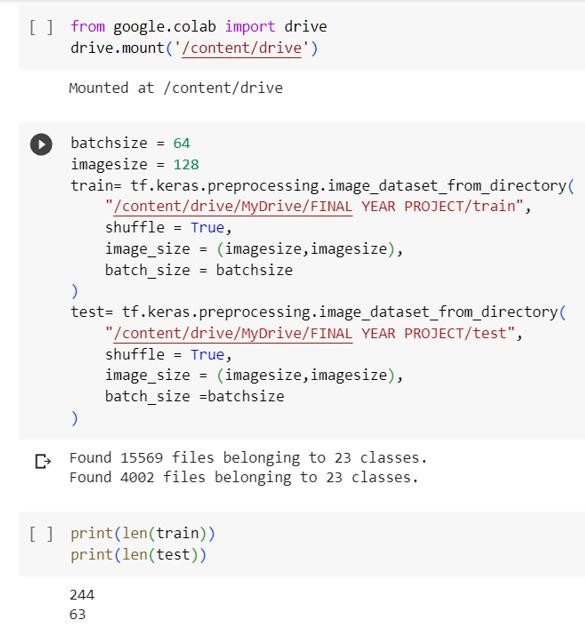
* + 1. **Sklearn: -** A free machine learning library for the Python programming language is called Scikit-learn (also known as sklearn). Support-vector machines, naive Bayes, decision trees, random forests, and k-means are among the classification, regression, and clustering methods included in it. Additionally, it offers resources for choosing and analyzing models and prepping data.

Scikit-learn is a popular choice for machine learning in Python

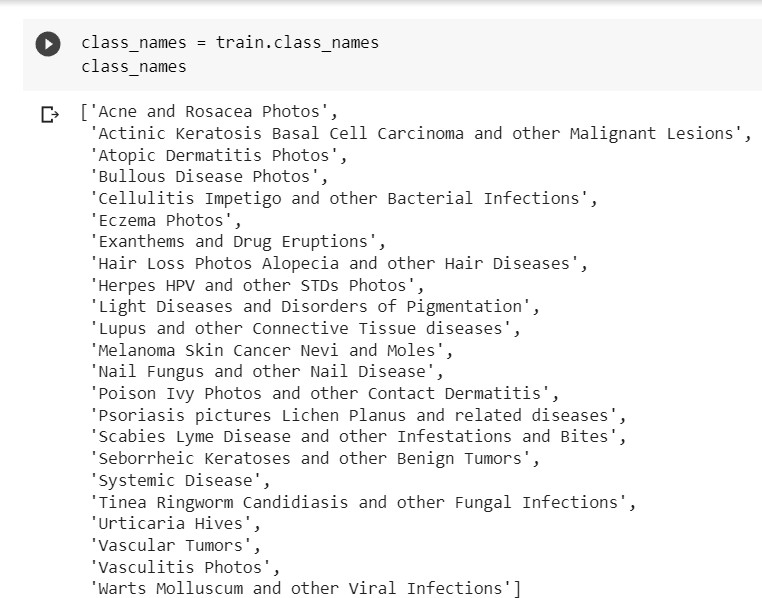
because it is easy to use, efficient, and has a wide range of features. It is also well-documented and has a large community of users and contributors.

A strong tool that may be used to address a variety of machine learning issues is scikit-learn. Scikit-learn is a good place to start if you are new to machine learning. Scikit-learn can be a useful tool for prototyping new concepts and for creating effective machine learning models if you are an experienced machine learning practitioner.

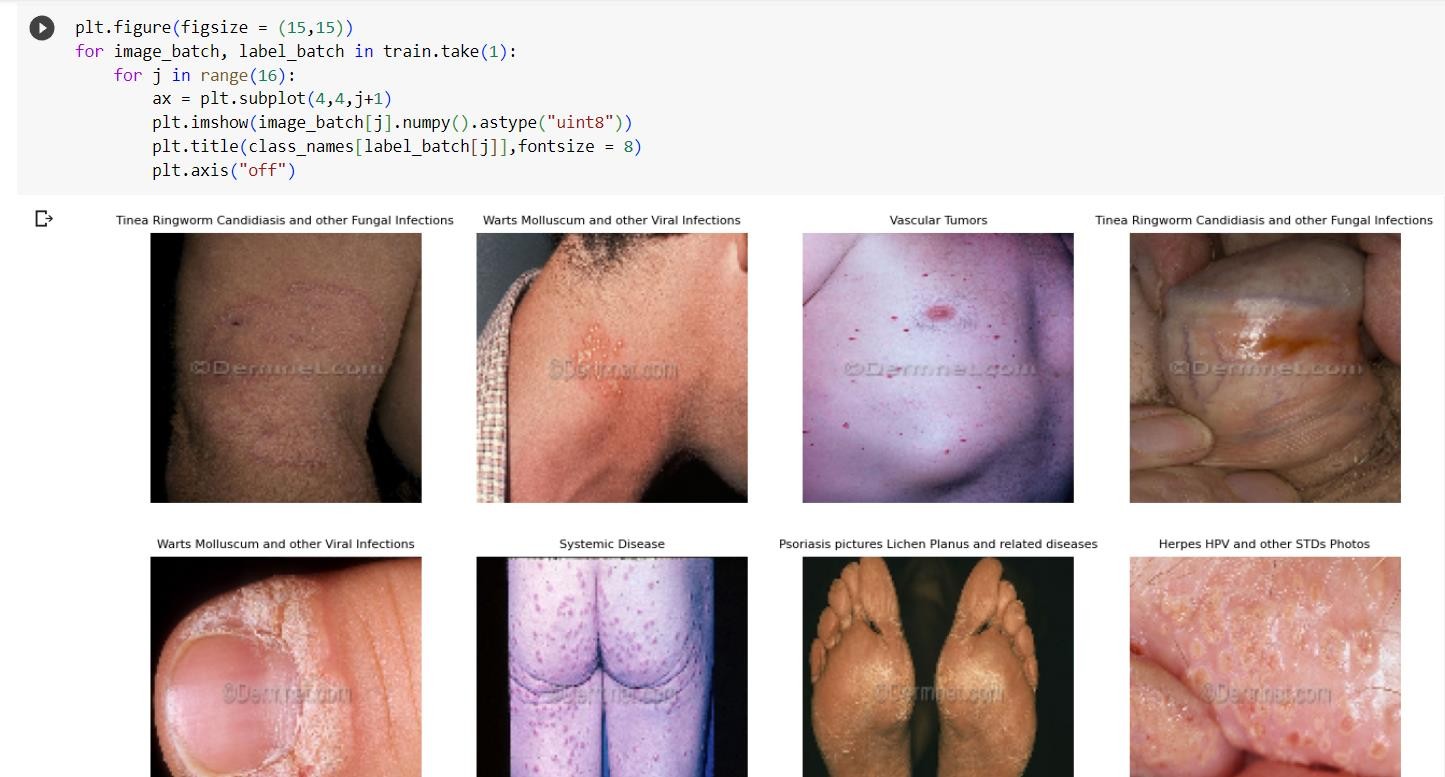
* 1. **Snapshots:**



**Fig.5.1 A**



**Fig.5.2 B**



###### Fig.5.3 C



**Fig.5.4 D**

## CHAPTER – 6

**TESTING**

In machine learning, Testing is the process of confirming that the models are performing as predicted. Given that machine learning models are frequently intricate and sensitive to data changes, it can be a difficult procedure.

A trained machine learning model's performance and accuracy are assessed using untrained data, which is the process known as testing in machine learning. It offers information about the model's predictive ability and aids in evaluating how effectively the model applies to fresh, untested data.

A training set and testing (or validation) set are often created from the available data in the machine learning testing process. The training set is used to develop the model, and the testing set is used to evaluate its performance.

## Manual Testing

A Quality Assurance Analyst will be physically carrying out testing on the goods. It is carried out to identify bugs in the code that is being developed. In manual testing, the examiner verifies all of the key features of the software under consideration. Without the use of any computerized programming testing instruments, the product analysts conduct the tests and write the test reports in this interaction. It is a tried-and-true method for finding problems in

programming frameworks across all testing kinds. To execute the product testing measure, an experienced analyst is typically in charge.

###### Functional Testing

It is a type of programming testing where the framework is examined in comparison to real-world requirements and specifics. Useful testing ensures that the application properly satisfies the requirements or conclusions. This type of testing is particularly concerned with the effects of planning. It emphasizes replicating actual framework usage but does not support any assumptions about the structure of the framework.

It may be essentially described as a type of testing that verifies that every feature of the product application operates in accordance with the necessity and specificity. The application's source code is not a concern for this testing. By providing appropriate test input, anticipating the yield, and comparing the real yield and the expected yield, every application of the product's usefulness is tested.

###### Non-Functional Testing

It is a type of programming testing done to ensure the application's non-utilitarian requirements are met. It verifies whether the framework is acting in accordance with the requirement or not. It examines all of the angles that utilitarian testing does not.

Programming testing that examines non-utilitarian components of

a product application is known as non-practical testing. Its goal is to evaluate the framework's development in light of non-functional constraints that utilitarian testing never addresses. Practical testing is important, but non-practical testing is also important.

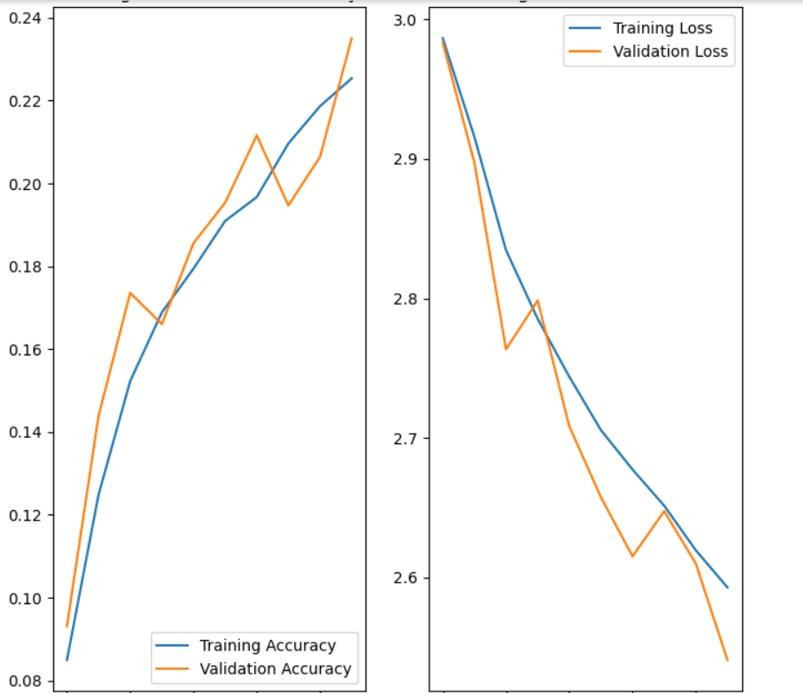
###### Test Cases Snapshot



**Fig.6.1 A**



###### Fig.6.2 B



**Fig.6.3 C**





**Fig.6.4 D**

## CHAPTER-7 CONCLUSION

1. A promising area of dermatology research is convolutional neural network (CNN)-based skin disease identification. A deep learning model called a CNN may be trained to examine photos and spot patterns in the skin.
2. Studies have demonstrated that CNNs are capable of correctly classifying various skin lesions, including seborrheic keratoses, basal cell carcinomas, and melanoma. Additionally, according to some research, CNNs are just as effective as dermatologists at spotting skin cancer.
3. It's crucial to keep in mind, though, that the application of CNNs to the identification of skin diseases is still in its infancy, and further research is required to raise the technology's accuracy and dependability. Additionally, a clinical examination and biopsy are required in addition to the use of CNNs in order to confirm the diagnosis.
4. All things considered, the application of CNNs has the potential to enhance the early detection and diagnosis of skin illnesses, particularly in areas with limited resources and restricted access to dermatologists. To guarantee accurate and trustworthy findings, the technology must be fully validated and integrated with clinical practice.

## CHAPTER – 8 LIMITATIONS AND FUTURE SCOPE

#### LIMITATION OF THE PROJECT

There are several limitations to skin disease detection using convolutional neuralnetworks (CNNs) which include:

**Limited dataset**: One of the major difficulties in applying CNNs to diagnose skin diseases is the dearth of high-quality, diverse, and well-annotated datasets. The lack of significant, diverse, and well- annotated datasets may be the root cause of models that do not generalize well to new images and perform poorly in real-world scenarios.

**Overfitting**: When the dataset is tiny, the excessive number of parameters in CNNs might cause overfitting. When a model performs well on training data but badly on test data, this is referred to as overfitting and shows that the model has learned training data rather than learning general features.

**Limited generalizability:** Age, sex, race, and skin type are among the many variables that might affect the presentation of skin diseases. Models may not be able to generalize successfully to new patients with varied characteristics if they were developed using a small dataset.

**Limited interpretability**: CNNs are intricate models, making it challenging to comprehend how the model generates its predictions. Because of this, it may be difficult for clinicians to believe and rely on the predictions made by the model.

**Lack of robustness:** CNNs can produce different predictions because they are sensitive to even little rotation, scale, and translation changes in the input pictures. Because of this, applying the model to situations where the image quality and lighting may differ in real-world settings can be difficult.

**Ethical and privacy issues**: Due to the collection, storage, and use of private medical information, the use of CNNs to diagnose skin diseases also poses ethical and privacy issues.

#### FUTURE SCOPE OF PROJECT

The future scope of skin disease detection using convolutional neural networks (CNNs) includes several potential areas of research:

**Improving the diversity and size of the dataset:** It is crucial to have a sizable and varied collection of photos that represents a wide range of skin types, ages, and races in order to enhance the generalizability of CNN models.

**Developing robust and interpretable models:** It is necessary to

create more reliable models that can accommodate changes in lighting, rotation, and scale as well as interpretable models that are simple for physicians to use.

**Incorporating clinical information:** The models' accuracy can be increased and their applicability in healthcare practices increased by incorporating clinical data such as patient demographics and medical history.

**Enhancing the performance of models on rare skin diseases:** there is less data available and it is harder to diagnose rare skin disorders, it is necessary to improve the performance of models.

**Developing multimodal models:** Creating multimodal models that can integrate input from several sources, including pictures, clinical data, and even genetic information, can result in predictions that are more precise and dependable.

**Implementing the technology in clinical practice:** Implementing the technology in real-world circumstances and comparing its performance to the present standard of care is crucial for making it more effective in clinical practice.

**Addressing ethical and privacy concerns:** The collecting, storage, and use of sensitive medical data are among the ethical and privacy issues that are brought up by the use of CNNs to diagnose skin diseases. It's critical to address these issues and make sure that technology is applied morally and sensibly.

In conclusion, the future of skin disease detection with CNNs encompasses a variety of research areas with the goal of enhancing the generalizability, interpretability, robustness, and clinical utility of the models, as well as resolving ethical and privacy issue.

## CHAPTER – 9 REFERENCES

* + 1. Hay, R. J., Johns, N. E., Williams, H. C., Bolliger, I. W., Dellavalle, R. P., Margolis, D. J., ... & Naghavi, M. (2014). The global burden of skin disease in 2010: an analysis of the prevalence and impact of skin conditions. *Journal of Investigative Dermatology*, *134*(6), 1527-1534.
    2. Huang, X., Zhang, J., Li, J., Zhao, S., Xiao, Y., Huang, Y., ... & Shen, M. (2019). Daily intake of soft drinks and moderate-to- severe acne vulgaris in Chinese adolescents. *The Journal of pediatrics*, *204*, 256-262.
    3. Deng, Y., Peng, Q., Yang, S., Jian, D., Wang, B., Huang, Y., ... & Li, J. (2018). The Rosacea-specific Quality-of-Life instrument (RosQol): revision and validation among Chinese patients. *PLoS One*, *13*(2), e0192487.
    4. Chen, J., Zeng, W., Pan, W., Peng, C., Zhang, J., Su, J., ... & Chen, X. (2018). Symptoms of systemic lupus erythematosus are diagnosed in leptin transgenic pigs. *PLoS Biology*, *16*(8), e2005354.
    5. Bewley, A. (2017). The neglected psychological aspects of skin

disease. *BMJ*, *358*.

* + 1. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248-255). Ieee.
    2. Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *nature*, *542*(7639), 115- 118.
    3. Zhang, X., Wang, S., Liu, J., & Tao, C. (2018). Towards improving diagnosis of skin diseases by combining deep neural network and human knowledge. *BMC medical informatics and decision making*, *18*(2), 69-76.
    4. Masood, A., & Ali Al-Jumaily, A. (2013). Computer aided diagnostic support system for skin cancer: a review of techniques and algorithms. *International journal of biomedical imaging*, *2013*.
    5. Gurovich, Y., Hanani, Y., Bar, O., Nadav, G., Fleischer, N., Gelbman, D., ... & Gripp, K. W. (2019). Identifying facial phenotypes of genetic disorders using deep learning. *Nature medicine*, *25*(1), 60-64.

\*\*\* SKIN DISEASE USING CNN \*\*\*

Number of Classes - 23 DATASET

1. Training Dataset - 15557 Files in JPG Format
2. Testing Dataset - 4002 Files in JPG Format *#Import necessary libraries and modules* **import** tensorflow **as** tf

**from** tensorflow.keras **import** layers

**from** tensorflow.keras.preprocessing.image **import** ImageDataGenerator

**from** keras.applications **import** VGG16

**from** keras.preprocessing.image **import** ImageDataGenerator

**from** sklearn.svm **import** SVC

**from** sklearn.metrics **import** accuracy\_score

*#Define the input image dimensions and the number of classes in your dataset*

img\_height = 224

img\_width = 224

num\_classes = 23

*#Create a data generator object to preprocess and augment your dataset*

data\_generator = ImageDataGenerator( rescale=1./255, shear\_range=0.2, zoom\_range=0.2, horizontal\_flip=True, validation\_split=0.2)

*#Drive Access*

**from** google.colab **import** drive drive.mount('/content/drive')

Mounted at /content/drive

*#Use the data generator object to load and split your dataset into training and validation sets*

train\_generator = data\_generator.flow\_from\_directory( '/content/drive/MyDrive/FINAL YEAR PROJECT/train', target\_size=(img\_height, img\_width), batch\_size=32,

class\_mode='categorical', subset='training') *# set as training data*

validation\_generator = data\_generator.flow\_from\_directory( '/content/drive/MyDrive/FINAL YEAR PROJECT/train', target\_size=(img\_height, img\_width), batch\_size=32,

class\_mode='categorical', subset='validation')

Found 12463 images belonging to 23 classes. Found 3106 images belonging to 23 classes.

*#Define a CNN model architecture using the tf.keras.Sequential API*

model = tf.keras.Sequential([

layers.Conv2D(32, (3, 3), activation='relu',

input\_shape=(img\_height, img\_width, 3)),

layers.MaxPooling2D(pool\_size=(2, 2)),

layers.Conv2D(64, (3, 3), activation='relu'),

layers.MaxPooling2D(pool\_size=(2, 2)), layers.Flatten(),

layers.Dense(128, activation='relu'), layers.Dense(num\_classes, activation='softmax')

])

*#Compile the model with appropriate loss and optimizer functions*

model.compile( loss='categorical\_crossentropy', optimizer='adam', metrics=['accuracy'])

*#Train the model on your dataset using the fit\_generator method*

history = model.fit\_generator( train\_generator,

steps\_per\_epoch=train\_generator.samples//train\_generator.batch\_size, epochs=10,

validation\_data=validation\_generator,

validation\_steps=validation\_generator.samples//validation\_generator.ba tch\_size)

<ipython-input-8-428efbb955d4>:2: UserWarning: `Model.fit\_generator` is deprecated and will be removed in a future version. Please use

`Model.fit`, which supports generators. history = model.fit\_generator(

Epoch 1/10

389/389 [==============================] - 2787s 7s/step - loss:

3.0546 - accuracy: 0.1417 - val\_loss: 2.9388 - val\_accuracy: 0.1186 Epoch 2/10

389/389 [==============================] - 1499s 4s/step - loss:

2.7614 - accuracy: 0.1789 - val\_loss: 2.8676 - val\_accuracy: 0.1698 Epoch 3/10

389/389 [==============================] - 1487s 4s/step - loss:

2.6651 - accuracy: 0.2096 - val\_loss: 2.8699 - val\_accuracy: 0.1827 Epoch 4/10

389/389 [==============================] - 1483s 4s/step - loss:

2.5933 - accuracy: 0.2292 - val\_loss: 2.8229 - val\_accuracy: 0.1814 Epoch 5/10

389/389 [==============================] - 1494s 4s/step - loss:

2.5294 - accuracy: 0.2492 - val\_loss: 2.8333 - val\_accuracy: 0.1885 Epoch 6/10

389/389 [==============================] - 1489s 4s/step - loss:

2.4814 - accuracy: 0.2593 - val\_loss: 2.8358 - val\_accuracy: 0.1840 Epoch 7/10

389/389 [==============================] - 1520s 4s/step - loss:

2.4225 - accuracy: 0.2723 - val\_loss: 2.8567 - val\_accuracy: 0.1843 Epoch 8/10

389/389 [==============================] - 1489s 4s/step - loss:

2.3589 - accuracy: 0.2980 - val\_loss: 2.8444 - val\_accuracy: 0.1756 Epoch 9/10

389/389 [==============================] - 1489s 4s/step - loss:

2.3183 - accuracy: 0.3141 - val\_loss: 2.9013 - val\_accuracy: 0.1898 Epoch 10/10

389/389 [==============================] - 1502s 4s/step - loss:

2.2681 - accuracy: 0.3257 - val\_loss: 2.9869 - val\_accuracy: 0.1985

*# Evaluate the model on the test set*

test\_generator = data\_generator.flow\_from\_directory( '/content/drive/MyDrive/FINAL YEAR PROJECT/test', target\_size=(img\_height, img\_width), batch\_size=32,

class\_mode='categorical')

scores = model.evaluate(test\_generator) print('Test loss:', scores[0]) print('Test accuracy:', scores[1])

Found 4002 images belonging to 23 classes.

126/126 [==============================] - 824s 7s/step - loss: 3.1559

- accuracy: 0.0315

Test loss: 3.155900478363037

Test accuracy: 0.03148425742983818

*# Load pre-trained CNN model*

model = VGG16(weights='imagenet', include\_top=False)

*# Extract features from images*

datagen = ImageDataGenerator(rescale=1./255) train\_generator = datagen.flow\_from\_directory(

'/content/drive/MyDrive/FINAL YEAR PROJECT/train', target\_size=(224, 224),

batch\_size=32, class\_mode=None, shuffle=False)

train\_features = model.predict(train\_generator) test\_generator = datagen.flow\_from\_directory(

'/content/drive/MyDrive/FINAL YEAR PROJECT/test',

target\_size=(224, 224), batch\_size=32, class\_mode=None, shuffle=False)

test\_features = model.predict(test\_generator)

*# Flatten features*

train\_features = train\_features.reshape(train\_features.shape[0], -1) test\_features = test\_features.reshape(test\_features.shape[0], -1)

*# Train SVC model*

svc = SVC(kernel='linear') svc.fit(train\_features, train\_generator.classes)

*# Test SVC model*

y\_pred = svc.predict(test\_features)

accuracy = accuracy\_score(test\_generator.classes, y\_pred) print("Accuracy:", accuracy)

Downloading data from https://storage.googleapis.com/tensorflow/keras- applications/vgg16/vgg16\_weights\_tf\_dim\_ordering\_tf\_kernels\_notop.h5 58889256/58889256 [==============================] - 0s 0us/step

Found 15569 images belonging to 23 classes.

487/487 [==============================] - 8915s 18s/step

Found 4002 images belonging to 23 classes.

126/126 [==============================] - 2267s 18s/step Accuracy: 0.392303848075962

*SKIN DISEASE USING CNN*

Number of Classes - 23 DATASET

Training Dataset - 15557 Files in JPG Format Testing Dataset - 4002 Files in JPG Format

**import** numpy **as** np

**import** tensorflow **as** tf

**from** tensorflow.keras **import** models, layers

**import** matplotlib.pyplot **as** plt

**from** google.colab **import** drive drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force\_remount=True).

batchsize = 32

imagesize = 128

train= tf.keras.preprocessing.image\_dataset\_from\_directory( "/content/drive/MyDrive/FINAL YEAR PROJECT/train", shuffle = True,

image\_size = (imagesize,imagesize), batch\_size = batchsize

)

test= tf.keras.preprocessing.image\_dataset\_from\_directory( "/content/drive/MyDrive/FINAL YEAR PROJECT/test", shuffle = True,

image\_size = (imagesize,imagesize), batch\_size =batchsize

)

Found 15569 files belonging to 23 classes. Found 4002 files belonging to 23 classes.

print(len(train)) print(len(test))

487

126

type(train) tensorflow.python.data.ops.batch\_op.\_BatchDataset class\_names = train.class\_names

class\_names

['Acne and Rosacea Photos',

'Actinic Keratosis Basal Cell Carcinoma and other Malignant Lesions', 'Atopic Dermatitis Photos',

'Bullous Disease Photos',

'Cellulitis Impetigo and other Bacterial Infections', 'Eczema Photos',

'Exanthems and Drug Eruptions',

'Hair Loss Photos Alopecia and other Hair Diseases', 'Herpes HPV and other STDs Photos',

'Light Diseases and Disorders of Pigmentation', 'Lupus and other Connective Tissue diseases', 'Melanoma Skin Cancer Nevi and Moles',

'Nail Fungus and other Nail Disease',

'Poison Ivy Photos and other Contact Dermatitis', 'Psoriasis pictures Lichen Planus and related diseases', 'Scabies Lyme Disease and other Infestations and Bites', 'Seborrheic Keratoses and other Benign Tumors', 'Systemic Disease',

'Tinea Ringworm Candidiasis and other Fungal Infections', 'Urticaria Hives',

'Vascular Tumors', 'Vasculitis Photos',

'Warts Molluscum and other Viral Infections']

plt.figure(figsize = (15,15))

**for** image\_batch, label\_batch **in** train.take(1):

**for** j **in** range(16):

ax = plt.subplot(4,4,j+1) plt.imshow(image\_batch[j].numpy().astype("uint8")) plt.title(class\_names[label\_batch[j]],fontsize = 8) plt.axis("off")



**def** dataset\_partitions(ds, train\_split = 0.9,val\_split = 0.1,shuffle = True,shuffle\_size =10000):

**assert**(train\_split+val\_split) == 1 ds\_size = len(ds)

**if** shuffle:

ds = ds.shuffle(shuffle\_size, seed = 24) train\_size = int(train\_split\*ds\_size) val\_size = int(val\_split \* ds\_size)

train\_ds = ds.take(train\_size)

val\_ds = ds.skip(train\_size).take(val\_size)

**return** train\_ds, val\_ds

train\_ds, val\_ds =dataset\_partitions(train) len(val\_ds)

48

train\_ds = train\_ds.cache().shuffle(1000).prefetch(buffer\_size=tf.data.AUTOTUNE) val\_ds\_ds = val\_ds.cache().shuffle(1000).prefetch(buffer\_size=tf.data.AUTOTUNE) test\_ds = test.cache().shuffle(1000).prefetch(buffer\_size=tf.data.AUTOTUNE)

resizerescale = tf.keras.Sequential([

layers.experimental.preprocessing.Resizing(imagesize,imagesize), layers.experimental.preprocessing.Rescaling(1.0/255)

])

augmentation = tf.keras.Sequential([

layers.experimental.preprocessing.RandomFlip("horizontal\_and\_vertical"

),

layers.experimental.preprocessing.RandomRotation(0.2),

])

train\_ds = train\_ds.map(

**lambda** x, y: (augmentation(x, training=True), y)

).prefetch(buffer\_size=tf.data.AUTOTUNE)

model = models.Sequential([

resizerescale,

layers.Conv2D(64, kernel\_size = (3,3), activation = 'relu', input\_shape = (batchsize,imagesize,imagesize,3) ),

layers.AveragePooling2D((2,2)), layers.Conv2D(128, kernel\_size = (3,3),

activation = 'relu'),

activation = 'relu'),

activation='relu'),

activation='relu'),

])

layers.AveragePooling2D((2,2)), layers.Conv2D(128, kernel\_size = (3,3),

layers.AveragePooling2D((2,2)), layers.Conv2D(128, (3, 3),

layers.AveragePooling2D((2, 2)),

layers.Conv2D(128, (3, 3),

layers.AveragePooling2D((2, 2)), layers.Flatten(),

layers.Dense(128, activation='relu'), layers.Dense(23, activation='softmax')

model.build(input\_shape = (batchsize,128,128,3)) model.summary()

Model: "sequential\_2"

|  |  |  |
| --- | --- | --- |
| Layer (type) | Output Shape | Param # |
| ================================================================= | | |
| sequential (Sequential) | (32, 128, 128, 3) | 0 |
| conv2d (Conv2D) | (32, 126, 126, 64) | 1792 |
| average\_pooling2d (AverageP ooling2D) | (32, 63, 63, 64) | 0 |
| conv2d\_1 (Conv2D) | (32, 61, 61, 128) | 73856 |
| average\_pooling2d\_1 (Averag ePooling2D) | (32, 30, 30, 128) | 0 |
| conv2d\_2 (Conv2D) | (32, 28, 28, 128) | 147584 |
| average\_pooling2d\_2 (Averag ePooling2D) | (32, 14, 14, 128) | 0 |
| conv2d\_3 (Conv2D) | (32, 12, 12, 128) | 147584 |
| average\_pooling2d\_3 (Averag ePooling2D) | (32, 6, 6, 128) | 0 |
| conv2d\_4 (Conv2D) | (32, 4, 4, 128) | 147584 |
| average\_pooling2d\_4 (Averag ePooling2D) | (32, 2, 2, 128) | 0 |
| flatten (Flatten) | (32, 512) | 0 |
| dense (Dense) | (32, 128) | 65664 |
| dense\_1 (Dense) | (32, 23) | 2967 |

=================================================================

Total params: 587,031

Trainable params: 587,031

Non-trainable params: 0

model.compile( optimizer='adam', loss =

tf.keras.losses.SparseCategoricalCrossentropy(from\_logits=False), metrics=['accuracy']

)

epoch = 10

history = model.fit( train\_ds,

epochs = epoch, batch\_size = batchsize, validation\_data = val\_ds, verbose = 1,

)

Epoch 1/10

438/438 [==============================] - 924s 2s/step - loss: 2.9651

* accuracy: 0.1009 - val\_loss: 2.9095 - val\_accuracy: 0.1296 Epoch 2/10

438/438 [==============================] - 864s 2s/step - loss: 2.8619

* accuracy: 0.1438 - val\_loss: 2.8227 - val\_accuracy: 0.1510 Epoch 3/10

438/438 [==============================] - 851s 2s/step - loss: 2.8091

* accuracy: 0.1677 - val\_loss: 2.8128 - val\_accuracy: 0.1479 Epoch 4/10

438/438 [==============================] - 851s 2s/step - loss: 2.7661

* accuracy: 0.1769 - val\_loss: 2.7693 - val\_accuracy: 0.1771 Epoch 5/10

438/438 [==============================] - 937s 2s/step - loss: 2.7279

* accuracy: 0.1887 - val\_loss: 2.7000 - val\_accuracy: 0.1849 Epoch 6/10

438/438 [==============================] - 925s 2s/step - loss: 2.6939

* accuracy: 0.1985 - val\_loss: 2.6606 - val\_accuracy: 0.2012 Epoch 7/10

438/438 [==============================] - 933s 2s/step - loss: 2.6637

* accuracy: 0.2062 - val\_loss: 2.6146 - val\_accuracy: 0.2116 Epoch 8/10

438/438 [==============================] - 934s 2s/step - loss: 2.6374

* accuracy: 0.2127 - val\_loss: 2.6282 - val\_accuracy: 0.2246 Epoch 9/10

438/438 [==============================] - 931s 2s/step - loss: 2.6042

* accuracy: 0.2235 - val\_loss: 2.6102 - val\_accuracy: 0.2337 Epoch 10/10

438/438 [==============================] - 913s 2s/step - loss: 2.5852

* accuracy: 0.2311 - val\_loss: 2.5461 - val\_accuracy: 0.2526

acc = history.history['accuracy']

val\_acc = history.history['val\_accuracy']

loss = history.history['loss'] val\_loss = history.history['val\_loss']

plt.figure(figsize=(8, 8))

plt.subplot(1, 2, 1)

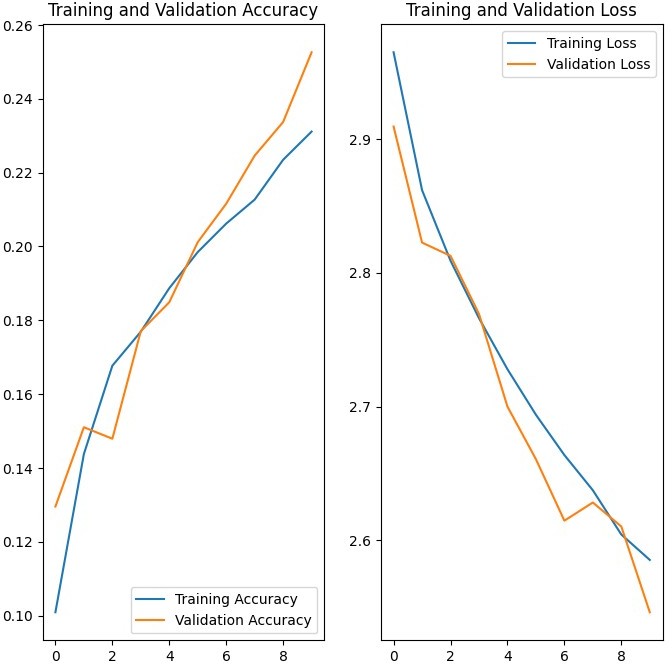
plt.plot(range(epoch), acc, label='Training Accuracy') plt.plot(range(epoch), val\_acc, label='Validation Accuracy') plt.legend(loc='lower right')

plt.title('Training and Validation Accuracy')

plt.subplot(1, 2, 2)

plt.plot(range(epoch), loss, label='Training Loss') plt.plot(range(epoch), val\_loss, label='Validation Loss') plt.legend(loc='upper right')

plt.title('Training and Validation Loss') plt.show()



**for** images\_batch, labels\_batch **in** test\_ds.take(1):

first\_image = images\_batch[23].numpy().astype('uint8') first\_label = labels\_batch[25].numpy()

print("first image to predict") plt.imshow(first\_image)

print("actual label:",class\_names[first\_label])

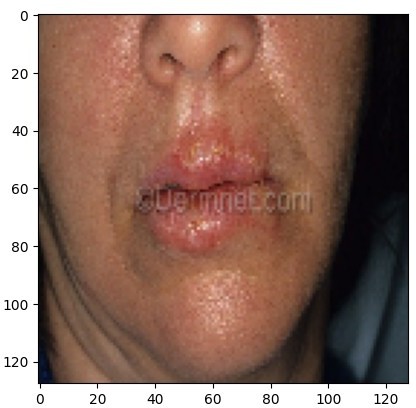
batch\_prediction = model.predict(images\_batch) print("predicted

label:",class\_names[np.argmax(batch\_prediction[23])])

first image to predict

actual label: Psoriasis pictures Lichen Planus and related diseases 1/1 [==============================] - 4s 4s/step

predicted label: Tinea Ringworm Candidiasis and other Fungal Infections



**def** predict(model, img): img\_array =

tf.keras.preprocessing.image.img\_to\_array(images[i].numpy()) img\_array = tf.expand\_dims(img\_array, 0)

predictions = model.predict(img\_array)

predicted\_class = class\_names[np.argmax(predictions[0])] confidence = round(100 \* (np.max(predictions[0])), 2) **return** predicted\_class, confidence

plt.figure(figsize=(15, 15))

**for** images, labels **in** test\_ds.take(1):

**for** i **in** range(9):

ax = plt.subplot(3, 3, i + 1)

plt.imshow(images[i].numpy().astype("uint8"))

predicted\_class, confidence = predict(model, images[i].numpy())

actual\_class = class\_names[labels[i]]

plt.title(f"Actual: {actual\_class},\n Predicted:

{predicted\_class}.\n Confidence: {confidence}%")

plt.axis("off")

1/1 [==============================] - 0s 241ms/step

1/1 [==============================] - 0s 39ms/step

1/1 [==============================] - 0s 37ms/step

1/1 [==============================] - 0s 43ms/step

1/1 [==============================] - 0s 42ms/step

1/1 [==============================] - 0s 39ms/step

1/1 [==============================] - 0s 37ms/step

1/1 [==============================] - 0s 36ms/step

1/1 [==============================] - 0s 36ms/step





**SPECIAL SECTION ON DATA-ENABLED INTELLIGENCE FOR DIGITAL HEALTH**

Received April 18, 2019, accepted May 15, 2019, date of publication May 22, 2019, date of current version June 4, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2918221*

Studies on Different CNN Algorithms for Face Skin Disease Classification

Based on Clinical Images

ZHE WU 1,5,6, SHUANG ZHAO2,3,4, YONGHONG PENG 7, XIAOYU HE1,5, XINYU ZHAO1,5, KAI HUANG2,3,4, XIAN WU6, WEI FAN6, FANGFANG LI2,3,4, MINGLIANG CHEN2,3,4,

JIE LI2,3,4, WEIHONG HUANG 2,5, XIANG CHEN2,3,4, AND YI LI1,5

1 School of Automation, Central South University, Changsha 410083, China

2 Department of Dermatology, Xiangya Hospital, Central South University, Changsha 410083, China

3 Hunan Key Laboratory of Skin Cancer and Psoriasis, Central South University, Changsha 410083, China

4 Hunan Engineering Research Center of Skin Health and Disease, Central South University, Changsha 410083, China

5 Mobile Health Ministry of Education-China Mobile Joint Laboratory, Central South University, Changsha 410083, China

6 Tencent Medical AI Lab, Beijing 100080, China

7 Faculty of Computer Science, University of Sunderland, Sunderland SR6 0DD, U.K.

Corresponding authors: Weihong Huang (whuangcn@qq.com), Xiang Chen (chenxiangck@126.com), and Yi Li (liyi1002@csu.edu.cn)

This work was supported in part by the National Key R&D Program of China under Grant 2018YFC0117000, and in part by the Special Foundation for Basic Science and Technology Research Program, Central South University, under Grant 2015FY111100.

**ABSTRACT** Skin problems not only injure physical health but also induce psychological problems, especially for patients whose faces have been damaged or even disfigured. Using smart devices, most of the people are able to obtain convenient clinical images of their face skin condition. On the other hand, the convolutional neural networks (CNNs) have achieved near or even better performance than human beings in the imaging field. Therefore, this paper studied different CNN algorithms for face skin disease classification based on the clinical images. First, from Xiangya–Derm, which is, to the best of our knowledge, China’s largest clinical image dataset of skin diseases, we established a dataset that contains 2656 face images belonging to six common skin diseases [seborrheic keratosis (SK), actinic keratosis (AK), rosacea (ROS), lupus erythematosus (LE), basal cell carcinoma (BCC), and squamous cell carcinoma (SCC)]. We performed studies using five mainstream network algorithms to classify these diseases in the dataset and compared the results. Then, we performed studies using an independent dataset of the same disease types, but from other body parts, to perform transfer learning on our models. Comparing the performances, the models that used transfer learning achieved a higher average precision and recall for almost all structures. In the test dataset, which included 388 facial images, the best model achieved 92.9%, 89.2%, and 84.3% recalls for the LE, BCC, and SK, respectively, and the mean recall and precision reached 77.0% and 70.8%.

**INDEX TERMS** Deep learning, CNN, facial skin disease, medical image processing.

1. **INTRODUCTION**

Based on a survey in 2010, skin diseases had the fourth leading cause of nonfatal disease burden in the world, and three of the world’s most common diseases were skin dis- eases [1]. Skin diseases have caused enormous economic burdens both in high-income and low-income countries. For each individual, skin problems can have adverse effects on all aspects of life, including interpersonal relationships, work, social functioning, physical activity and mental health.

The associate editor coordinating the review of this manuscript and approving it for publication was Linbo Qing.

Usually, skin diseases cause skin lesions, scales, plaques, pigmentation and other symptoms on the patient’s skin [2]–[5]. These symptoms result in long-term pain and disfigurement. Such damage not only injures physical health but also contribute to serious mental problems, especially when such damage occurs on face. Studies [6]–[8] showed that patients with primary skin diseases (such as psoriasis, alopecia areata and vitiligo) have a higher potential for mental problems, such as anxiety and depression. In addition, some skin disease treatments also have the possibility of inducing mental illness (such as isotretinoin, an acne medication, may induce suicidal depression).

VOLUME 7, 2019

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.

Personal use is also permitted, but republication/redistribution requires IEEE permission.

See <http://www.ieee.org/publications_standards/publications/rights/index.html>for more information.

66505

Z. Wu *et al.*: Studies on Different CNN Algorithms for Face Skin Disease Classification Based on Clinical Images

Facial skin is exposed to the air almost all the time, so it has a higher risk of being damaged than other areas. Moreover, facial skin is the most important part of the body for people’s appearance, so people are more concerned about their facial skin health than skin health anywhere else.

Along with the availability of massive amounts data brought by the Internet [9] and the improvement of com- puting power brought by advanced hardware, deep learning algorithms have achieved human-level performance in many fields. For example, convolutional neural networks (CNNs) have made many breakthroughs in the field of medical image processing, especially for pathological, CT and MRI images, which have rigid features and high resolution. However, research on clinical images is relatively insufficient. For these reasons, clinical images always contain a very complex con- text, and it is hard to control the conditions of acquiring the image. These circumstances make image processing tasks difficult.

Furthermore, datasets of a certain part of the body, espe- cially the face, re relatively scarce. At present, most of the available datasets are not clearly labeled with information on the body parts; for some datasets that provide this informa- tion, the proportion of facial images is always small [10]. All of these conditions make research difficult.

Therefore, this paper first constructed a skin image dataset based on 6 common facial skin diseases (seborrheic keratosis (SK), actinic keratosis (AK), rosacea (ROS), lupus erythe- matosus (LE), basal cell carcinoma (BCC), and squamous cell carcinoma (SCC)). It includes 2,656 facial images for a total of 4,394 images. We focus on these diseases for the following reasons: 1) LE, ROS, BCC and SCC frequently occur on the face; 2) AK and SK usually transition from benign to malignant without timely treatment.

Based on the dataset, experiments were carried out on 5 different CNN structures to verify whether these methods can effectively diagnose facial skin diseases using clinical images. In the test set consisting entirely of facial images, the structure named Inception-ResNet-v2 achieved the high- est average precision (77.0%).

1. **RELATED WORKS**

Many studies have applied deep learning algorithms to skin diseases [10]–[12]. For example, the performance in the task of classifying skin tumors using the Inception-v3 network has reached the level of professional dermatologists; for nine classes of tumors, a computer achieved an accuracy of 55.4%, and two dermatologists achieved accuracies of 53.3% and 55.0% [10]. Using the same network structure, [11] achieved an accuracy of 87.25 2.24% on the dermoscopic images for four common skin diseases, including SK, BCC, psoria- sis and melanocytic nevus. These studies show that current deep learning methods have the potential to be applied to dermatoses.

±

At the same time, the application of deep learning to face- related diseases is also promising. Reference [13] designed a deep learning algorithm called DeepGestalt and trained their

model on more than 17,000 real facial images of genetic syndromes, and this model can identify more than 200 genetic syndromes using facial images with relatively high precision. Reference [14] investigated using CNNs to classify acne into different severity grades ranging from clear to severe, and their results show that the accuracy of their method outperformed expert physicians.

Initially, we investigated the proportion of facial images in the most commonly used public datasets for skin dis- ease, which include AtlasDerm [15], DermlS [16], the ISIC Archive [17], Derm101 [18] and Dermnet [19]. Most of these datasets [15]–[18] did not provide information about body parts. In [19], which does provide body parts infor- mation, there were only 195 facial images. It is difficult to perform further research on facial skin diseases using such limited data. As a result, building a specialized dataset for face images is extremely necessary for our research.

1. **FACE IMAGE DATASET**

First, this paper established a dataset based on facial skin disease images, including 6 common skin diseases. The images in the dataset were obtained from Xiangya-Derm. These images and labels were rigorously reviewed by at least three experienced dermatologists. It will be made public after relevant procedures are completed.

Xiangya-Derm consists of 150,223 clinical images from 543 different skin diseases. Each image is captured by digital camera and has a matched pathology and medical history. This construct was produced by the Department of Dermatol- ogy, Xiangya School of Medicine, Central South University. To the best of our knowledge, it is the largest clinical image dataset of skin disease for computer-aided diagnosis (CADx). The details of the data distribution are shown in Table 1 and Figure 1. It is worth mentioning that the training set and the test set are divided according to different patients, which means that images of the same patient are prevented from

appearing in both the training set and the test set.

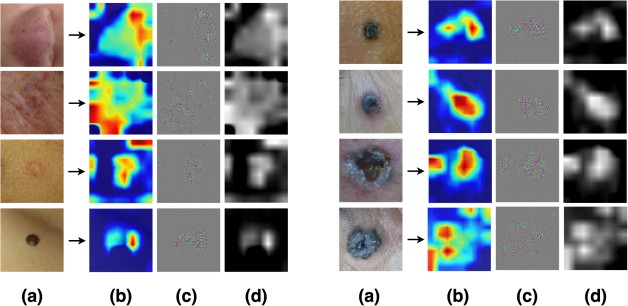
Our study was approved by the Ethics Committee of Xiangya School of Medicine, Central South University, and all participants provided informed consent.

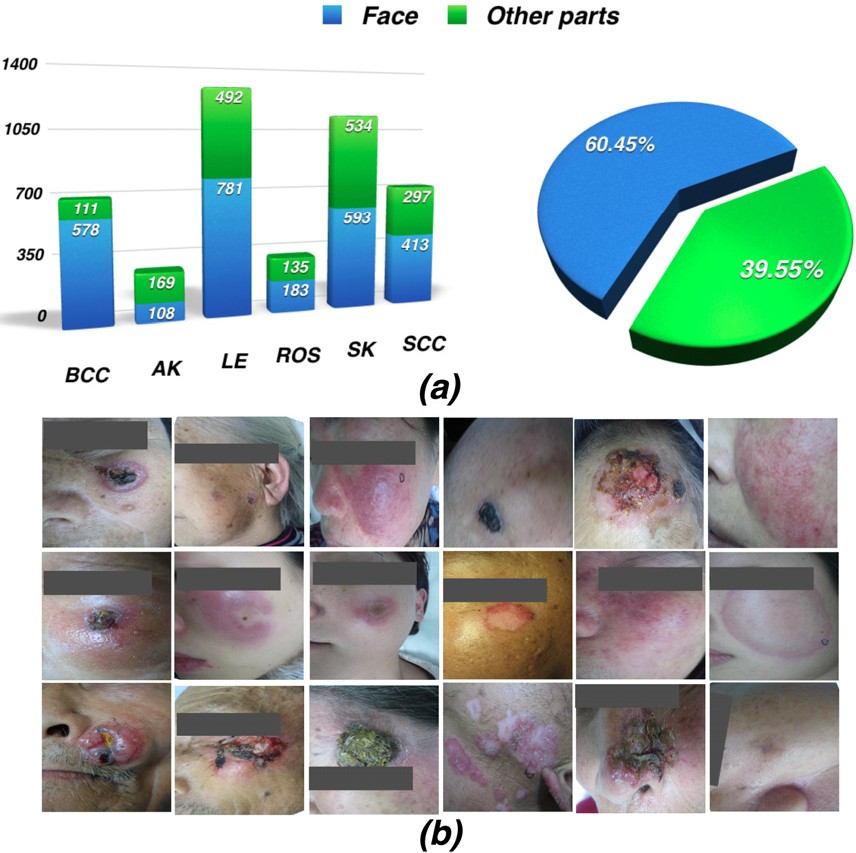
1. **METHODS**

A neural network is a mathematical model inspired by the transfer process of biological neuron information, and its purpose is to learn a mapping from input to output. By using a loss function as a constraint and backpropagation to optimize the parameters, this method can automatically learn complex tasks for different fields. This method has reduced the need for human labor, such as manual feature extraction and data reconstruction for classification. A CNN is a type of neural network. It generally consists of an input layer, many hidden convolutional layers, and an output layer. Using this struc- ture, the model can include a large number of parameters and obtain some usable properties, such as equivariance, for image-related tasks.

66506 VOLUME 7, 2019

Z. Wu *et al.*: Studies on Different CNN Algorithms for Face Skin Disease Classification Based on Clinical Images



**FIGURE 1.** The stacked cylinder diagram ((a) left) and the pie chart ((a) right) show the distribution of the images of face and other body

parts for different diseases in the dataset; (b) shows some examples of the dataset.

**TABLE 1.** Summary of dataset.



In this paper, we used five mainstream CNN algorithms that have been pretrained on ImageNet [9]. These five struc- tures include ResNet-50, Inception-v3, DenseNet121, Xcep- tion and Inception-ResNet-v2. We used same pre-process for these images, including random reverse and crop. And to address the problem of data imbalance, we used different weights in the cost function for different diseases.

ResNet adds connections between the shallow and deep layers of the network. Such connections directly transmit the information of the shallow layer to the deep layer. On the other hand, the propagation of the gradient to the shallow layer during backpropagation greatly increases the number of network layers [20].

The basic module of the Inception structure is the inception block. There are different kernels in a block, and each type of kernel has a different shape; the output of the block is combines the output from different kernels. This improves the diversity of the network in terms of width and the diver- sity of the scale of the receptive field. Therefore, the model improved its recognition performance for objects with differ- ent sizes [21], [22].

**FIGURE 2.** (a) is the input images, (b) is the heatmap for the combination of input images and the corresponding output of grad-CAM(d). (c) is the output of guided propagation.

DenseNet adds connections between each two layers; that is, the output feature maps of each layer will be used as the input for all subsequent layers. Using these dense connec- tions, the network reuses features, thereby improving perfor- mance with fewer parameters, which makes the calculation more efficient [23].

Xception is an updated version of the Inception structure. Xception improves the Inception module with a depthwise separable convolution. This change decouples spatial correla- tions and cross-channel correlations. It can obtain a better per- formance than Inception-v3 with the same parameters [24].

To some extent, Inception-ResNet is a combination of Inception and ResNet structures. By adding a residual con- nection to the Inception network, it can train deeper networks while maintaining the scale diversity of the network, thereby enhancing the performance [25].

In this paper, we used the same 300∗300 input images for each network and did not change the basic structure from that

in their origin paper. We replaced the first fully connected layer behind the last convolutional layer with global aver- age pooling and a 1∗1 convolution to reduce the number of parameters and maintain spatial information. Finally, we used a 1024-d fully connected layer in each network and then

used a softmax or logistic regression classifier to obtain 6 confidence outputs for six facial skin diseases. More details about the model structures are shown in Table 2, where *incep- tion block, dense block, transition layer,*and*inception resnet block*are modules that are the same as those from the origin papers.

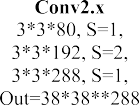
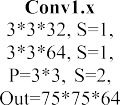
1. **RESULTS AND DISCUSSIONS**

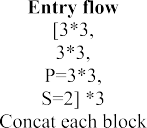
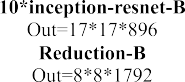
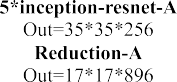
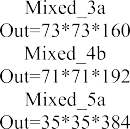
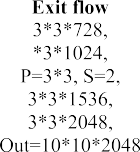
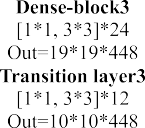
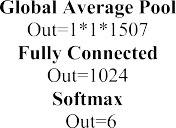
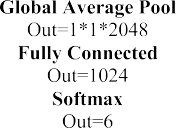
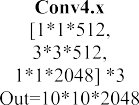
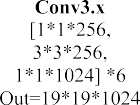
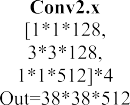
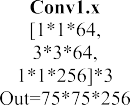
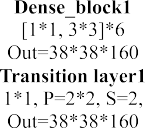
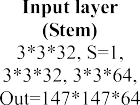
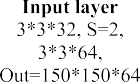
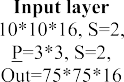
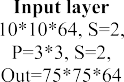
First, the five models are trained using only clinical facial images. The results obtained are shown in Table 3. The Inception-ResNet-v2 structure achieved the highest perfor- mance. Then, we pretrained the model with the data of other body parts and used the parameters from the pretrained model as the initial parameters for the new model, which is a method called transfer learning. The results are shown in Table 4.

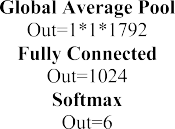
Comparing the results of Tables 3 and 4, the performance of the models that were pretrained on other body part images are generally superior to those models that were trained using only the facial images. In our opinion, there are so many differences between the images of different parts of the body.

VOLUME 7, 2019 66507

Z. Wu *et al.*: Studies on Different CNN Algorithms for Face Skin Disease Classification Based on Clinical Images

**TABLE 2.** Structures of the five models.



**TABLE 3.** Results of the model which have not been pre-trained by images of other body parts.







However, for most diseases, the difference in symptoms on different body parts is not obvious. Therefore, when training the model for skin diseases, a better strategy would be to use the data of the whole body to train the model and then use it as the initialization and retrain model on images of a



particular body part. Then, the model can be used to diagnose the disease at that specific body part.

As shown in Table 3 and Table 4, among the five network structures, the Inception-ResNet-v2 structure achieved a bet- ter performance. However, the recall for AK is only 54.1%.

Z. Wu *et al.*: Studies on Different CNN Algorithms for Face Skin Disease Classification Based on Clinical Images

**TABLE 4.** Results of the model which has been pre-trained by images of other body parts.



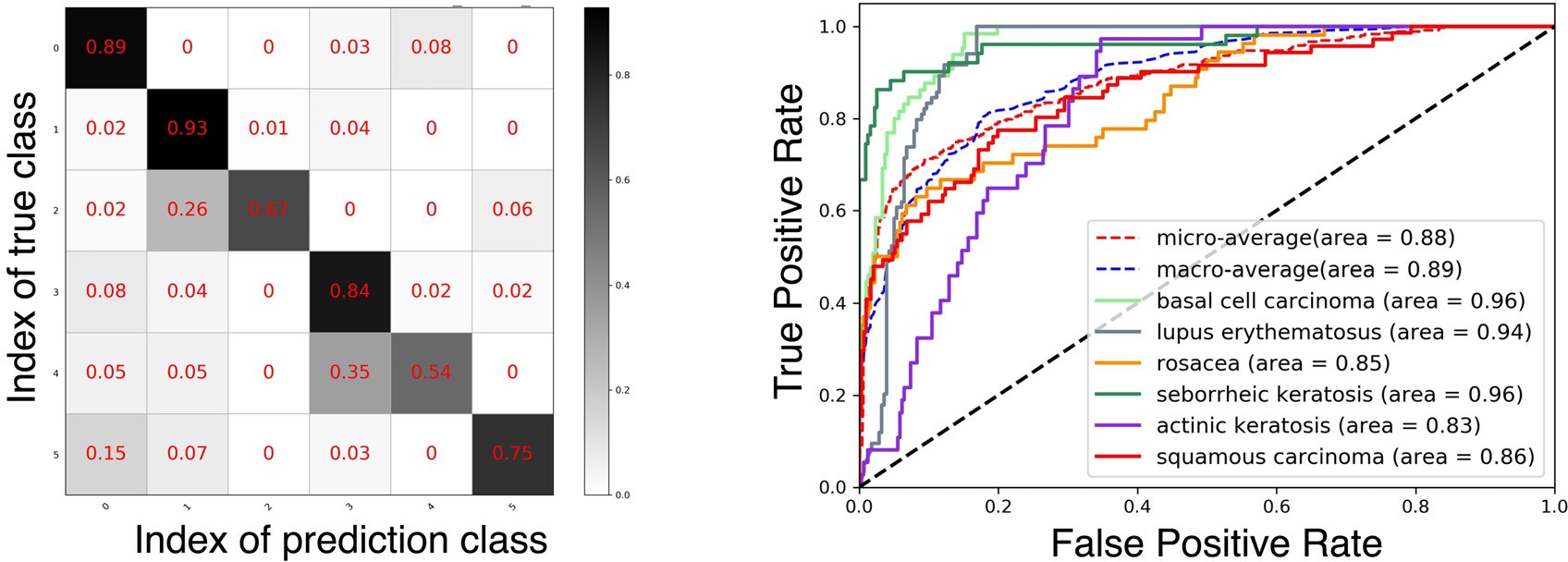
       







**FIGURE 3.** Confusion matrix (left) and ROC curve (right) of the Inception-ResNet v2 model on the test set. Class 0-5 are corresponding to BCC, LE, ROS, SK, AK, SCC respectively.

By analyzing the confusion matrix (Figure 3), we found that most of the misjudgment samples of AK were classified as SK. After consulting experts, we believe that the primary rea- son is the similar clinical manifestations between SK and AK, which include brown rashes, flat papules and local keratoses. Another possible reason is the imbalance in the number of samples for these two diseases. In the dataset, the number of SK images is significantly more than the number of AK images. These two reasons combined cause misclassifica- tions. As a result, we believe that at such an insufficient data scale, it is hard for current CNN structures to determine the difference between such similar diseases. This issue should

be studied in future studies.

Furthermore, a drawback of deep learning is that the out- put is hard to explain. To explore what the networks have learned from these images indeed, we used grad-CAM [26] and guided propagation [27] to visualize the output results of Inception-ResNet-v2. For a particular output class, areas that contribute more to classification than other areas are shown in warmer colors in the heatmap. The results [Figure 2] show that, generally, CNNs indeed made their predictions by using features learned from lesion areas or other areas with

abnormalities, rather than from the textures or other features of normal skin.

1. **CONCLUSION**

This paper performed experiments using five mainstream CNN structures for the clinical image diagnosis of six com- mon facial skin diseases and constructed a data set consisting mainly of facial skin disease images. The results demon- strate that CNNs have the ability to recognize facial skin diseases. Based on our experiments, we determined that dif- ferent models to diagnose diseases on different body parts should be used. Furthermore, our experiments also showed that a more reasonable network structure could improve the performance of the model. The performance of the current network structure has been satisfactory in some diseases, but the overall performance has yet to be improved. As a result, if we want that people to actually use this technique to check their face skin health in their daily life, specialized improvements should be developed.

In our opinion, the application of artificial intelligence techniques in the medical field is not sufficient, and the datasets from this field should be improved both in quantity

Z. Wu *et al.*: Studies on Different CNN Algorithms for Face Skin Disease Classification Based on Clinical Images

and quality. With the increasing amount of facial image data of various skin diseases and the continuous improvement of the network structure, CNN-based facial skin disease diag- nosis algorithms will continue to improve in performance. We believe that, in the future, patients will use convenient CNN-based applications to keep their face skin healthy.

**ACKNOWLEDGMENT**

A lot of support has been provided by doctors and nurses in the dermatology department of Xiangya Hospital, Central South University. *(Zhe Wu and Shuang Zhao contributed equally to this work.)*

**REFERENCES**

1. R. J. Hay, N. E. Johns, H. C. Williams, I. W. Bolliger, R. P. Dellavalle, and

D. J. Margolis, ‘‘The global burden of skin disease in 2010: An analysis of the prevalence and impact of skin conditions,’’ *J. Investigative Dermatol- ogy*, vol. 134, no. 6, pp. 1527–1534, 2014.

1. X. Huang, J. Zhang, J. Li, S. Zhao, Y. Xiao, Y. Huang, D. Jing, L. Chen,

X. Zhang, J. Su, Y. Kuang, W. Zhu, M. Chen, X. Chen, and M. Shen, ‘‘Daily intake of soft drinks and moderate-to-severe acne vulgaris in Chinese Adolescents,’’ *J. Pediatrics*, vol. 204, pp. 256–262, Jan. 2018.

1. Y. Deng, Q. Peng, S. Yang, D. Jian, B. Wang, Y. Huang, H. Xie, and J. Li, ‘‘The rosacea-specific quality-of-life instrument (RosQol): Revision and validation among Chinese patients,’’ *PLoS ONE*, vol. 13, no. 2, Feb. 2018, Art. no. e0192487.
2. C. Junchen, W. Zeng, W. Pan, C. Peng, J. Zhang, J. Su, W. Long,

H. Zhao, X. Zuo, X. Xie, J. Wu, L. Nie, H.-Y. Zhao, H.-J. Wei, and

X. Chen, ‘‘Symptoms of systemic lupus erythematosus are diagnosed in leptin transgenic pigs,’’ *PLoS Biol.*, vol. 16, no. 8, Aug. 2018, Art. no. e2005354.

1. X. Xiaoyun, H. Chaofei, Z. Weiqi, C. Chen, L. Lixia, L. Queping, P. Cong,

Z. Shuang, S. Juan, and C. Xiang, ‘‘Possible involvement of F1F0-ATP synthase and intracellular ATP in Keratinocyte differentiation in normal skin and skin lesions,’’ *Sci. Rep.*, vol. 7, Feb. 2017, Art. no. 42672.

1. A. Bewley, ‘‘The neglected psychological aspects of skin disease,’’ *Brit. Med. J.*, vol. 358, p. 3208, Jul. 2017.
2. W. Chen, X. Zhang, W. Zhang, C. Peng, W. Zhu, and X. Chen, ‘‘Polymor- phisms of SLCO1B1 rs4149056 and SLC22A1 rs2282143 are associated with responsiveness to acitretin in psoriasis patients,’’ *Sci. Rep.*. vol. 4, no. 1, 2018, Art. no. 13182. doi: [10.1038/s41598-018-31352-2.](http://dx.doi.org/10.1038/s41598-018-31352-2)
3. X. Zhou, W. Zhu, M. Shen, Y. He, C. Peng, Y. Kuang, J. Su, S. Zhao,

X. Chen, and W. Chen, ‘‘Frizzled-related proteins 4 (SFRP4) rs1802073G allele predicts the elevated serum lipid levels during acitretin treatment in psoriatic patients from Hunan, China,’’ *PeerJ*, vol. 13, no. 6, 2018, Art. no. e4637.

1. J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. F. Fei, ‘‘ImageNet: A large-scale hierarchical image database,’’ in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
2. A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and

S. Thrun, ‘‘Dermatologist-level classification of skin cancer with deep neural networks,’’ *Nature*, vol. 542, pp. 115–118, Feb. 2017.

1. X. Zhang, S. Wang, J. Liu, and C. Tao, ‘‘Towards improving diagnosis of skin diseases by combining deep neural network and human knowledge,’’ *Med. Inform. Decis. Making*, vol. 18, no. 2, p. 59, 2018.
2. A. Masood and A. A. Al-Jumaily, ‘‘Computer aided diagnostic support system for skin cancer: A review of techniques and algorithms,’’ *Int. J. Biomed. Imag.*, vol. 2013, pp. 1–22, 2013.
3. Y. Gurovich, Y. Hanani, O. Bar, G. Nadav, N. Fleischer, D. Gelbman,

L. Basel-Salmon, P. M. Krawitz, S. B. Kamphausen, M. Zenker, L. M. Bird, and K. W. Gripp, ‘‘Identifying facial phenotypes of genetic disorders using deep learning,’’ *Nature Med.*, vol. 25, pp. 60–64, Jan. 2019.

1. A. Melina, N. N. Dinh, B. Tafuri, G. Schipani, S. Nisticò, C. Cosentino,

F. Amato, D. Thiboutot, and A. Cherubini, ‘‘Artificial intelligence for the objective evaluation of acne investigator global assessment,’’ *J. Drugs Dermatology*, vol. 17, no. 9, pp. 1006–1009, 2018.

1. A. Dermatology. *Samuel Freire Da Silva, Delso Bringel Calheiros*. [Online]. Available: [http://www.atlasdermatologico.com.br](http://www.atlasdermatologico.com.br/)
2. DermIS.net. *The Dept. of Clinical Social Medicine (Univ. of Heidelberg) and the Dept. of Dermatology (Univ. of Erlangen)*. Accessed: Apr. 2019. [Online]. Available: [http://www.dermis.net](http://www.dermis.net/)
3. *ISDIS Treasurer, Angel Cummings, Aadi Kalloo, University DermatologyCenter*. Accessed: Apr. 2019. [Online]. Available: [https://www.isicarchi](http://www.isicarchive.com/%23!/topWithHead-er/onlyHeaderTop/gallery)v[e.com/#!/topWithHead-er/onlyHeaderT](http://www.isicarchive.com/%23!/topWithHead-er/onlyHeaderTop/gallery)op/g[allery](http://www.isicarchive.com/%23!/topWithHead-er/onlyHeaderTop/gallery)
4. *Derm101*. [Online]. Available: [https://www.derm101.com/imagelibrary/?](http://www.derm101.com/imagelibrary/) match=IN
5. *Dermnet, Thomas Habif*. [Online]. Available: [http://www.dermnet.com](http://www.dermnet.com/)
6. K. He *et al.*, ‘‘Deep residual learning for image recognition,’’ 2015.
7. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan,

V. Vanhoucke, A. Rabinovich, ‘‘Going deeper with convolutions,’’ in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

1. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, ‘‘Rethinking the inception architecture for computer vision,’’ in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
2. G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, ‘‘Densely connected convolutional networks,’’ in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4700–4708.
3. F. Chollet, ‘‘Xception: Deep learning with depthwise separable convolu- tions,’’ in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2016, pp. 1251–1258.
4. C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, ‘‘Inception-v4, Inception-ResNet and the impact of residual connections on learning,’’ in *Proc. 31st AAAI Conf. Artif. Intell.*, Feb. 2017, pp. 1–7.
5. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and

D. Batra, ‘‘Grad-CAM: Visual explanations from deep networks via gradient-based localization,’’ in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 2016, pp. 618–626.

1. J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, ‘‘Striving for simplicity: The all convolutional net,’’ Dec. 2014, *arXiv:1412.6806*. [Online]. Available: https://arxiv.org/abs/1412.6806

ZHE WU was born in Changsha, Hunan, China, in 1995. He received the B.E. degree in automa- tion from the Nanjing Institute of Technology. He is currently pursuing the M.S. degree in control science and engineering (artificial intelligence and pattern recognition) with the School of Automa- tion, Central South University, Changsha.

Since 2016, he has been with the Laboratory of Intelligent Systems and Intelligent Control, Cen- tral South University. Since 2019, he has been an

Intern Researcher with the Tencent Medical AI Lab, Beijing. His recent work is about skin disease image, such as psoriasis, acne, and lupus. His research interests include machine learning, deep learning, computer vision, object detection, and semantic segmentation of medical image.

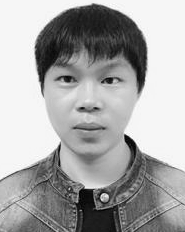
SHUANG ZHAO was born in Sichuan, China, in 1985. He received the M.D. and Ph.D. degrees from the Xiangya School of Medicine, Central South University, China, in 2013.

From 2015 to 2016, he was in his Postgraduate Training of Mohs surgery at The Third Affiliated Hospital, Zhongshan University, and the System- atic Training on skin surgery and photodynamic therapy at the Institute of Dermatology and Hospi- tal for Skin Diseases, Chinese Academy of Medi-

cal Sciences, and the Peking Union Medical College. Since 2015, he has been a member of the Standing Committee, Medical Aesthetics and Cosmetology Group, Chinese Medical Association. He is currently an Attending Doctor and an Assistant Research Fellow with the Department of Dermatology, Xiangya Hospital, Central South University. He has authored nine publica- tions. His research interests include surgical treatment of skin neoplasm, such as malignant melanoma, and surgery combined with photodynamic therapy to terat some refractory dermatosis, such as VIN and CIN.

Z. Wu *et al.*: Studies on Different CNN Algorithms for Face Skin Disease Classification Based on Clinical Images

YONGHONG PENG is currently with the Faculty of Computer Science, University of Sunderland, Sunderland, U.K.

XIAOYU HE received the B.E. degree in automa- tion from the Central South University of China, in 2016, where he is currently pursuing the M.S. degree in automation with the School of Informa- tion Science and Engineering. His research inter- ests include pattern recognition, deep learning, and medical image processing.

XINYU ZHAO is currently with the School of Automation, Central South University, Changsha, China.

KAI HUANG is currently with the Department of Dermatology, XiangYa Hospital, Central South University, Changsha, China.

XIAN WU is currently with the Tencent Medical AI Lab, Beijing, China.

WEI FAN is currently with the Tencent Medical AI Lab, Beijing, China.

FANGFANG LI is currently with the Department of Dermatology, XiangYa Hospital, Central South University, Changsha, China.

MINGLIANG CHEN is currently with the Department of Dermatology, XiangYa Hospital, Central South University, Changsha, China.

JIE LI is currently with the Department of Dermatology, XiangYa Hospital, Central South University, Changsha, China.

WEIHONG HUANG received the B.Eng. degree in automation and the M.Eng. degree in pattern recognition and smart control from Southeast Uni- versity, China, in 1995 and 1998, respectively, and the Ph.D. degree in computer science from Nan- jing University, China, in 2001.

From 2001 to 2002, he was a Postdoctoral Research Fellow with CNRS University Lyon 1 France. From 2002 to 2005, he was a Lecturer with the Department of Computer Science, Uni-

versity of Hull, U.K. From 2005 to 2014, he was a Senior Lecturer with the School of Computer and Information Systems, Kingston University London, U.K. In 2016, he was appointed as a Professor and the Depute Director of the Mobile Health Ministry of Education-China Mobile Joint Laboratory, Xiangya Hospital, Central South University, China. His research interests include mobile health, artificial intelligence in healthcare, cognitive computing for healthcare, semantic multimedia computing, and knowledge graph applications.

Dr. Huang is a Committee Member of the China Hospital Information Management Association, a Standing Committee Member of the Medical and Health Big Data Evaluation and Assurance Board of the Chinese Health Information and Big Data Association, and the Chairman of the Special- ized Committee of Information Management of Hunan Health Management Association.

XIANG CHEN received the B.M. and M.M. degrees from the Xiangya School of Medicine, Central South University, China, in 1988 and 1997, respectively, and the Ph.D. degree in dermatology from the School of Medicine, Kagoshima Univer- sity, Japan.

He is currently a Professor with the Derma- tology Department, Xiangya Hospital, Central South University. He is also the Chief Scientist of the Chinese Medical Association of Dermatology

Cancer Research Center. Meanwhile, he has served as the Chairman of the Committee of Hunan Society of Dermatology. He is also the Cheung Kong Scholars Program Distinguished Professor and the Gainer of the China National Funds for Distinguished Young Scientists. He has presided over 30 funds (including nine from the National Natural Science Foundation of China) and published over 80 SCI papers. His main research interests include skin cancers and psoriasis. He is also with the Chinese Society of Dermatology, the Dermatological Committee of Hunan Medical Doctor Association, the Youth Scientific Association of Central South University, and the Pharmacogenomics Committee of Chinese Pharmacological Society.

YI LI received the B.S. degree from The College of Information Engineering of Xiangtan Univer- sity, Xiangtan, in 2002, and the M.S. and Ph.D. degrees from the School of Automation, Central South University, Changsha, China, in 2005 and 2013, respectively, where he is currently a Lecturer with the Department of Automation Science and Technology. His research interests include span computer vision, machine learning, and pattern recognition.

### Project report ID06

ORIGINALITY REPORT

21

### %

SIMILARITY INDEX

16%

INTERNET SOURCES

10%

PUBLICATIONS

15%

STUDENT PAPERS

PRIMARY SOURCES

towardsdatascience.com

1

2

Internet Source %

# [www.researchgate.net](http://www.researchgate.net/)

2

2

Internet Source %

# my.clevelandclinic.org

3

2

Internet Source %

# [www.coursehero.com](http://www.coursehero.com/)

4

1

Internet Source %

# ijarsct.co.in

5

1

Internet Source %

# [www.warse.org](http://www.warse.org/)

6

1

Internet Source %

# dspace.cusat.ac.in

7

1

Internet Source %

# Submitted to Rajkia Engineering College

8

1

Student Paper %

# Submitted to British University in Egypt

9

1

Student Paper %

10

11

12

13

14

15

16

# uk.mathworks.com

Internet Source

# M. Paolanti, R. Pierdicca, M. Martini, A. Felicetti, E. S. Malinverni, E. Frontoni, P. Zingaretti. "DEEP CONVOLUTIONAL NEURAL NETWORKS FOR SENTIMENT ANALYSIS OF CULTURAL HERITAGE", ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2019

Publication

# vitalﬂux.com

Internet Source

# Submitted to Institute of Technology Carlow

Student Paper

# Submitted to Melbourne Institute of Technology

Student Paper

# T. Shanthi, R.S. Sabeenian, R. Anand. "Automatic Diagnosis of Skin Diseases using Convolution Neural Network", Microprocessors and Microsystems, 2020

Publication

# Submitted to Higher Ed Holdings

Student Paper

1%

1%

1%

1%

<1%

<1%

<1%

17

18

19

20

21

22

23

24

25

# "Neural Information Processing", Springer Science and Business Media LLC, 2017

Publication

# Submitted to Coventry University

Student Paper

# pdfcoﬀee.com

Internet Source

# umpir.ump.edu.my

Internet Source

# Benjamin Cohen, Adam Cadesky, Shuchie Jaggi. "Dermatologic manifestations of thyroid disease: a literature review", Frontiers in Endocrinology, 2023

Publication

# Submitted to University of Southern Queensland

Student Paper

# Submitted to Uttar Pradesh Technical University

Student Paper

# Submitted to European University of Luxembourg

Student Paper

# Submitted to University of Surrey

Student Paper

# Submitted to Vels University

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

26

27

28

29

30

31

32

33

34

35

Student Paper

# Submitted to Üsküdar Üniversitesi

Student Paper

# Submitted to Nettur Technical Training Foundation

Student Paper

# Submitted to Sim University

Student Paper

# text-id.123dok.com

Internet Source

# [www.bartleby.com](http://www.bartleby.com/)

Internet Source

# Submitted to London Metropolitan University

Student Paper

# Srigiri Krishnapriya, Yepuganti Karuna. "Pre- trained deep learning models for brain MRI image classiﬁcation", Frontiers in Human Neuroscience, 2023

Publication

# Submitted to Universiti Tunku Abdul Rahman

Student Paper

# Submitted to University of Strathclyde

Student Paper

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

36

37

38

39

40

41

42

43

44

# Tirthajyoti Sarkar. "Productive and Eﬃcient Data Science with Python", Springer Science and Business Media LLC, 2022

Publication

# 1library.net

Internet Source

# Submitted to Associatie K.U.Leuven

Student Paper

# Submitted to Technological Institute of the Philippines

Student Paper

# Submitted to Universidade Nova De Lisboa

Student Paper

# [www.mdpi.com](http://www.mdpi.com/)

Internet Source

# libcat.nshealth.ca

Internet Source

# mountainscholar.org

Internet Source

# Håvard Kvamme, Nikolai Sellereite, Kjersti Aas, Steﬀen Sjursen. "Predicting Mortgage Default using Convolutional Neural Networks", Expert Systems with Applications, 2018

Publication

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

45

46

47

48

49

50

51

52

53

# docslide.net

Internet Source

# [www.nature.com](http://www.nature.com/)

Internet Source

# "Machine Learning in Medical Imaging", Springer Science and Business Media LLC, 2019

Publication

# archive.interconf.center

Internet Source

# cupdf.com

Internet Source

# etd.lib.metu.edu.tr

Internet Source

# repository.hanyang.ac.kr

Internet Source

# [www.haikudeck.com](http://www.haikudeck.com/)

Internet Source

# [www.intechopen.com](http://www.intechopen.com/)

Internet Source

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

Exclude quotes On Exclude bibliography On

Exclude matches < 3 words