Computer-aided Approach for Multi Class Skin Diseases Classification Based On Skin Images NetworksUsing Convolutional Neural

Project report submitted to Visvesvaraya National

Institute of Technology, Nagpur in partial fulfillment

of the requirements for the award of the degree

Bachelor of Technology

In

Computer Science and Engineering

by

BT19CSE008 Apoorva Kumar

BT19CSE017 Shruti Eknath Borse

BT19CSE028 Eshan Kumar Jain

BT19CSE100 Sharma Ojas Rajnish

under the guidance of

Dr. S. R. Sathe

Department of Computer Science and Engineering Visvesvaraya National Institute of Technology Nagpur 440 010 (India)

2023

Computer-aided Approach for Multi Class Skin Diseases Classification Based On Skin Images Using Convolutional Neural Networks

Project report submitted to Visvesvaraya National Institute of Technology, Nagpur in partial fulfillment of the requirements for the award of the degree

Bachelor of Technology

In

Computer Science and Engineering

by

BT19CSE008 Apoorva Kumar

BT19CSE017 Shruti Eknath Borse

BT19CSE028 Eshan Kumar Jain

BT19CSE100 Sharma Ojas Rajnish

under the guidance of

Dr. S. R. Sathe

Department of Computer Science and Engineering Visvesvaraya National Institute of Technology Nagpur 440 010 (India)

2023

Department of Computer Science and Engineering

Visvesvaraya National Institute of Technology, Nagpur

Declaration

We, hereby declare that this project work titled **“Computer-aided Approach for Multi Class Skin Diseases Classification Based On Skin Images Using Convolutional Neural Networks**” is carried out by us in the **Department of Computer Science and Engineering** of Visvesvaraya National Institute of Technology, Nagpur. The work is original and has not been submitted earlier whole or in part for the award of any degree/diploma at this or any other Institution / University.

Apoorva Kumar
(BT19CSE008)

Shruti Eknath Borse
(BT19CSE017)

Eshan Kumar Jain
(BT19CSE028)

Sharma Ojas Rajnish
(BT19CSE100)

Department of Computer Science and Engineering

Visvesvaraya National Institute of Technology, Nagpur

CERTIFICATE

*This to certify that the project titled "**Computer-aided Approach for Multi Class Skin Diseases Classification Based On Skin Images Using Convolutional Neural Networks**", submitted by **Apoorva kumar, Shruti Eknath Borse, Eshan Kumar Jain, Sharma Ojas Rajnish** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering**, VNIT Nagpur. The work is comprehensive, complete and fit for final evaluation.*

Dr. P. S. Deshpande
Head,
Department of Computer Science and
Engineering, VNIT, Nagpur.

Dr. S. R. Sathe
Professor
Department of Computer Science and
Engineering,VNIT, Nagpur.

Date:

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our guide, **Dr. S. R. Sathe**, for giving us the opportunity to work on this research project of Comparative analysis, guiding us thoughtfully throughout this project and giving us directions whenever required. We would also like to thank **Sridhar Reddy Gogu** for his guidance and help in the implementation of the project and for providing the necessary resources whenever required by us.

We would like to thank all the teaching and non-teaching faculty of the Computer Science and Engineering Department for supporting us and providing the necessary facilities at all times. We would like to express our gratitude for availing college resources which enabled us to swiftly complete our project.

We would like to acknowledge the contributions of those who have constantly supported and encouraged us which helped in the successful completion of the project.

ABSTRACT

Skin diseases can have long lasting effects on a person's physical as well as mental health. As of now, the general trend for skin disease diagnosis is just the visual inspection by the dermatologist. The approaches for automatic detection or diagnosis have been emerging in recent years. But, most, if not all of the approaches do not have a very good performance, hence are not reliable.Therefore, there is a need to develop this project and model in order to make further progress in the field of early detection of various skin diseases. The shortcomings of existing skin disease classification systems are reviewed, and a new methodology is suggested. A new approach to identify skin diseases using segmentation and Convolutional Neural Networks (CNN) is proposed. To construct the dataset, we combined certain datasets and ensured a balanced distribution of skin disease samples. The dataset was augmented using ImageDataGenerator and a new segmentation methodology was developed to segment the images within the dataset. Various machine learning models were studied, and the most appropriate approach was chosen. The authors used an average ensemble of MobileNetV2 and ShuffleNet to achieve a testing accuracy of 93.2%, which outperformed the respective models chosen separately. Specifically, MobileNet achieved a testing accuracy of 91.25%, ShuffleNet achieved a testing accuracy of 78.14%, and VGG19 achieved a testing accuracy of 91.2%. This proposed approach shows promise in accurately identifying skin diseases, which can have significant implications in the medical field.

LIST OF FIGURES

LIST OF TABLES

INDEX

CHAPTER 1

INTRODUCTION

Skin diseases can range from minor conditions such as acne and eczema to more serious illnesses such as monkeypox. Detecting these diseases at an early stage is critical as it can have a significant impact on the outcome of treatment and the long-term health of the individual.It is essential to regularly monitor the skin for any changes in color, texture, or appearance and seek medical attention if any abnormalities are noticed. Early detection of skin diseases can improve the chances of successful treatment and a better long-term outcome.

1.1 Symptoms

There are various symptoms depending on skin diseases. Mostly these skin diseases are visual which are identified by dermatologists. For example,in Basal cell carcinoma disease patients can experience small, shiny bumps on skin and a pink and red patch of skin that may be itchy or tender which can be visually detected. Actinic keratosis is a precancerous condition that can potentially develop into skin cancer if left untreated. It is important to have any suspicious skin lesions evaluated by a medical professional.

1.2 Assessment

Assessing skin diseases using images and machine learning techniques such as Convolutional Neural Networks (CNNs) is becoming increasingly popular. Using these techniques, the system can extract the affected lesion region and identify the probable skin disease that the patient can have. To ensure the accuracy and reliability of the assessment, it is recommended to have a qualified dermatologist review the results and confirm the diagnosis.

CHAPTER 2

PROBLEM STATEMENT

Developing an automatic, objective and non-invasive multiple skin diseases recognition system using deep learning techniques to help in the monitoring and diagnosis of multiple skin diseases.

2.1 Objective

- Critically examines and compare performances of the existing deep learning (DL)-based methods to detect multiple skin diseases —focusing on Acne, actinic-keratosis, basal-cell-carcinoma, chickenpox, cowpox, dermatitis, eczema, erythema-multiforme, granuloma, herpes, monkeypox,psoriasis etc. —from skin images.
- Create a novel dataset of multiple skin diseases.
- Development of an automatic system for detecting multiple skin diseases by using data from clinical images and patient information using deep learning.

2.2 Challenges

Our first challenge was to build a large enough dataset covering various disease images set to perform preprocessing on it. In previous works till now they have used less dataset size and less diseases were covered. If the dataset used for training the CNN model is not diverse enough, the model may not be able to recognize certain skin diseases or may make incorrect predictions. It is essential to include images of different skin types, ages, and ethnicities to ensure that the model can recognize skin diseases in different populations. CNN models require large amounts of data to be trained effectively. However, collecting large, high-quality datasets can be challenging, especially for rare or uncommon skin diseases. Overfitting occurs when the CNN model is too complex and learns to recognize the training data too well, resulting in poor generalization to new data. Overfitting can be reduced by using techniques such as data augmentation, regularization, and early stopping. Some skin diseases are more prevalent than others, resulting in imbalanced datasets. This can lead to models that are biased towards more common skin diseases, resulting in lower accuracy for rare skin diseases.

CHAPTER 3

LITERATURE SURVEY

3.1 IMAGE DATA COLLECTION AND IMPLEMENTATION OF DEEP LEARNING-BASED MODEL IN DETECTING MONKEYPOX DISEASE USING MODIFIED VGG16[1]

This paper deals with the lack of a dataset containing Monkeypox infected. here they have created the dataset by collecting images from multiple open-source and online portals. They proposed and evaluated a modified VGG16 model, which includes two distinct studies: Study One and Two. Our exploratory computational results indicate that our suggested model can identify Monkeypox patients with an accuracy of 97 ± 1.8% (AUC = 97.2) and 88 ± 0.8% (AUC = 0.867) for Study One and Two, respectively.[1]

Proposed model: VGG16

Dataset: 4 classes

Performance:

Study	Training accuracy	Testing accuracy
Study 1	97.1%	83.1%
Study 2	88.1%	78.1%

Table 3.1 Performance analysis of paper [1]

Study	Label	Train set	Test set	Total
Study One	Monkeypox	34	9	43
	Chickenpox	38	9	47
	Total	72	18	90
Study Two	Monkeypox augmented	470	117	587
	Others	933	234	1167
	Total	1403	351	1754

Table 3.2 dataset distribution in paper[1]

Limitations observed in this paper:

- Small sample size
- Less number of diseases covered
- Overfitting observed

3.2 Monkeypox Skin Lesion Detection Using Deep Learning Models: A Feasibility Study[2]

In this paper, the study was about feasibility in the area of detection of monkeypox disease. The classification was of binary type(monkeypox vs others).
Amidst the ongoing efforts to recover from the impact of COVID-19, the Monkeypox virus has emerged as a potential global pandemic. While the virus itself is not as lethal or as transmissible as COVID-19, numerous new cases are being documented globally every day. This highlights the need to take adequate preventative measures, failing which, the world may have to confront another catastrophic pandemic. Recent progress in machine learning (ML) has shown tremendous potential in facilitating image-based diagnoses, such as detecting cancer, identifying tumor cells, and diagnosing COVID-19 patients. In light of this, we propose employing a similar approach to diagnose Monkeypox-related diseases, given that they manifest as skin conditions that can be captured using imaging techniques. To facilitate this, we are introducing the "Monkeypox2022" dataset, which is now freely available to the public via our GitHub repository. This dataset comprises images that we have collected from multiple open-source and online portals that allow unrestricted use, including for commercial purposes. This provides a safer and more democratized avenue for sharing data for the development and deployment of new ML models. Furthermore, we are pleased to present our new VGG16 model, which features two studies, namely Study One and Two. Our preliminary computational findings demonstrate that our suggested model can identify Monkeypox patients with a high degree of accuracy, achieving a score of 97±1.8% (AUC=97.2) and 88±0.8% (AUC=0.867) for Study One and Two, respectively. In addition, we have employed Local Interpretable Model-Agnostic Explanations (LIME) to explain our model's predictions and feature extraction which provide deeper insight to the distinctive features characterizing the onset of Monkeypox virus.

Performance Comparisons:

Proposed Models	Dataset Used	Performance
VGG16	MSLD	Accuracy: 81.48%
ResNet50	MSLD	Accuracy: 82.96%
InceptionV3	MSLD	Accuracy: 74.07%

Ensemble	MSLD	Accuracy: 79.26%

Table 3.3 Performance comparison of different models in paper[2]

Dataset Description: MLSD (available on kaggle):

DISTRIBUTION OF THE MONKEYPOX SKIN LESION DATASET (MSLD)

Class label	No. of Original Images	No. of Unique Patients	No. of Augmented Images
Monkeypox	102	55	1428
Others	126	107	1764
Total	228	162	3192

Table 3.4 Distribution of MSLD dataset

Limitations observed in this paper:
The dataset size is too low for real world applications
Binary classification also limits the real world usage of this system, because it can only be used for monkeypox detection.

3.3 A Model for Classification and Diagnosis of Skin Disease using Machine Learning and Image Processing Techniques[3]

This paper presents a model that takes an image of the skin affected by a disease and diagnoses acne, cherry angioma, melanoma, and psoriasis. The proposed model is composed of five steps, i.e., image acquisition, preprocessing, segmentation, feature extraction, and classification. In addition to using the machine learning algorithms for evaluating the model, i.e., Support Vector Machine (SVM), Random Forest (RF), and KNearest Neighbor (K-NN) classifiers, and achieved 90.7%,84.2%, and 67.1%, respectively[3]. Also, the SVM classifier result of the proposed model was compared with other papers, and mostly the proposed model's result is better. In contrast, one paper achieved an accuracy of 100%. This paper Highlighted the importance of preprocessing techniques in the field Otsu, texture extraction(glcm,ngtdm,gabor), edge detection(sobel) .

Skin diseases are a prevailing type of ailment that can be caused by a variety of factors such as fungal infection, bacteria, allergies, and viruses. While the development of lasers and photonics-based medical technology has allowed for more efficient and accurate diagnosis of skin diseases, the cost of said diagnosis remains limited and expensive. Recognizing this, image processing techniques have been utilized to create an automated screening system for dermatology in its preliminary stages. Feature extraction is a crucial component in the process of accurately identifying and classifying skin diseases, aided by various computer vision techniques. Saudi Arabia's desert climate and hot weather contribute to the high incidence of skin diseases in the region. Accordingly, this study proposes a simple, cost-effective image processing-based method for detecting skin diseases. Our approach utilizes digital images of the affected skin area, employs image analysis to classify the disease type, and generates the results in the form of disease type, spread, and severity. Our system employs a pretrained convolutional neural network to extract features from resized, colored images, and then utilizes multiclass SVM to classify the identified features. Our study has successfully achieved a 100% accuracy rate in detecting three types of skin diseases.

Ref	Disease	Segmentation Technique	Feature techniques	Accuracy
Hameed N et al. [5]	Acne, Psoriasis, Melanoma, Benign, Eczema, and Healthy Skin.	Otsu's	GLCM, NGTDM, and color spaces	83%
Sinthura S. et al. [6]	Acne, Psoriasis, Melanoma, and Rosacea.	Otsu's	GLCM	89%
Hameed N et al. [10]	Acne, Melanoma, Benign, Eczema, and Healthy Skin.		CNN: AlexNet.	86.21%
Proposed Model	Acne, Cherry angioma, Melanoma, and Psoriasis.	Otsu's	Entropy, Gabor, and Sobel.	90.7%

Table 3.5 Performance comparison of different models in paper[3]

Some limitations from this paper:
- Multi classification problem
- Dataset is still small

3.4 Skin Disease detection based on different Segmentation Techniques[4]

The skin, being the largest organ in the human body, serves as an outer layer of protection. Despite its importance, skin varies in type and pigmentation among individuals, providing an environment for different microorganisms to thrive. To prevent skin cancer, melanin is produced by melanocytes to absorb harmful UV radiation from the sun. Unfortunately, early identification of skin conditions is limited in third-world communities due to lack of resources. To address this, image segmentation techniques such as adaptive thresholding, edge detection, K-means clustering, and morphology-based segmentation can be employed to detect skin diseases and aid in early identification. Pre-processing steps such as deblurring and noise reduction are applied before analyzing specific patterns unique to each type of ailment to accurately identify the disease. In this study, segmentation techniques were evaluated on the basis of Signal to Noise Ratio and demonstrated promising results for detecting chickenpox, eczema, psoriasis, and ringworm. However, combining the segmentation procedure with disease classification could enhance efficiency and provide support for clinicians and dermatologists.

3.5 Detection and Classification of Skin diseases using Deep Learning[5]

For This paper, the proposed framework involves a deep learning-based method to detect skin diseases. This system will utilize computational techniques to analyze, process, and relegate the image data predicated on various features of the images. Dataset-Warts Mollusca, Systemic Disease, Seborrheic Keratosis, Nevus, Bullous, Actinic Keratosis, Acne and Rosacea etc.

Deep learning models are efficient in learning the features that assist in understanding complex patterns precisely. This study proposed a computerized process of classifying skin disease through deep learning based MobileNet V2 and Long Short Term Memory (LSTM). The MobileNet V2 model proved to be efficient with a better accuracy that can work on lightweight computational devices. The proposed model is efficient in maintaining stateful information for precise predictions. A gray-level co-occurrence matrix is used for assessing the progress of diseased growth. The performance has been compared against other state-of-the-art models such as Fine-Tuned Neural Networks (FTNN), Convolutional Neural Network (CNN), Very Deep Convolutional Networks for Large-Scale Image Recognition developed by Visual Geometry Group (VGG), and convolutional neural network architecture that expanded with few changes. The HAM10000 dataset is used and the proposed method has outperformed other methods with more than 85% accuracy. Its robustness in recognizing the affected region much faster with almost 2× lesser computations than the conventional MobileNet model results in minimal computational efforts. Furthermore, a mobile application is designed for instant and proper action. It helps the patient and dermatologists identify the type of disease from the affected region's image at the initial stage of the skin disease. These findings suggest that the proposed system can help general practitioners efficiently and effectively diagnose skin conditions, thereby reducing further complications and morbidity

CHAPTER 4

IMPLEMENTATION RESOURCES

Python

Python is the gold standard for solving Machine Learning / Deep Learning problems. It offers rich extensibility via the use of open-source libraries. In our work, we utilize **numpy** (for vector computations), **pandas** (for statistical operations), **OpenCV** (for image operations) and **keras** (for high-level ML/DL endpoints).

CHAPTER 5

DATASET

Our goal is to detect correct disease by performing deep learning techniques on images. For this we collected a large dataset of images of the skin diseases of interest. The dataset should be diverse and representative of the different variations and severities of the skin diseases. Perform preprocessing on these images and then Train a CNN model on the dataset. The below given dataset was combined from various different dataset and then the image samples for each disease were made uniform.[6][7][8][9]

5.1 Features

Constructed the required dataset for the problem which consists of 23 different classes of skin disease(22 diseases + healthy skin) which consists of different symptoms or legion region patterns /patterns on skin.

Here are the common visible symptoms of the skin diseases that we have included in our custom dataset:

1. Acne**:**
- Blackheads, whiteheads, or pimples on the face, neck, chest, or back.
- Oily skin.
- Inflamed or tender areas of the skin.

2. Actinic keratosis**:**
- Dry, scaly, or rough patches of skin.
- A growth or lesion that feels rough or gritty.
- Discolored or thickened skin in the affected area.

3. Basal cell carcinoma**:**
- A small, shiny bump on the skin.
- A pink or red patch of skin that may be itchy or tender.

4. Chickenpox**:**
- Red, itchy bumps that eventually turn into blisters.

5. Cowpox**:**

- Cowpox can cause skin lesions that appear as raised, red bumps that may be accompanied by fever and fatigue.

6. Dermatitis**:**
- Itchy, red, and swollen skin.
- Skin rash that can be dry, scaly, or blistered.
- Crusty, oozing, or weeping areas of the skin.
7. Eczema**:**
- Dry, itchy, and red skin.
- Rash that can appear on the face, neck, hands, feet, and inner elbows and knees.
- Cracked, scaly, or rough patches of skin.

8. Erythema Multiforme**:**
- Erythema Multiforme causes skin lesions that are often in the shape of a target or bull's eye, with a raised, red border and a central blister. These lesions can appear on the face, trunk, and extremities.

9. Granuloma**:**
- Granulomas are small, firm, and red or brown bumps that can develop on the skin. They are often painless and can occur in clusters.

10. Herpes**:**
- Painful blisters or sores on the mouth, genitals, or rectum.

11. Hidradenitis Suppurativa:
- Hidradenitis Suppurativa causes painful, deep-seated nodules or abscesses that can rupture and leak pus. These lesions can occur in the armpits, groin, buttocks, and other areas of the body where sweat glands are located.

12. Lupus**:**
- Butterfly-shaped rash on the face.

13. Measles: Measles causes a characteristic rash that starts on the face and spreads to the trunk and extremities. The rash appears as flat, red spots that may merge together.

14. Melanocytic Nevi**:** Melanocytic Nevi, also known as moles, are benign growths on the skin that can be flat or raised and may be brown, black, or flesh-colored.

15. Melanoma: Melanoma is a type of skin cancer that can appear as a new, unusual growth or as an existing mole that changes in size, shape, or color. Melanomas may have irregular borders and may be multicolored, with shades of brown, black, and red.

16. Molluscum: Molluscum causes small, raised, round, and smooth lesions on the skin that are often flesh-colored.

17. Monkeypox: Monkeypox can cause skin lesions that are similar to those of smallpox, including fever, fatigue, and a rash that starts on the face and spreads to the trunk and extremities.

18. Psoriasis:
- Red, scaly, and itchy patches of skin.
- Thick, silvery scales on the affected area.
- Dry, cracked skin that may bleed.

19. Rosacea:
- Redness on the face, typically on the cheeks, nose, chin, or forehead.
- Small, visible blood vessels on the face.
- Pimples or bumps on the face.

20. Tinea:
- Red, itchy, and scaly patches of skin.
- A rash that may be raised and have a distinct border.
- Peeling or cracking of the skin.

21. Vasculitis:
- Vasculitis can cause red or purple spots or bumps on the skin that may be painful or itchy. These lesions may appear in different shapes and sizes.

22. Warts:
- Small, rough bumps on the skin.
- Raised, flat, or cauliflower-like growths,tenderness.

5.2 Dataset Description

Dataset Name	**TAG**
Monkeypox Skin Lesion Dataset[6]	1
Skin diseases image dataset[7]	2
A Web-scraped Skin Image Database of Monkeypox, Chickenpox, Smallpox, Cowpox, and Measles[8]	3
Dermnet[9]	4

Table 5.1 Assigning tags to datasets

Unaugmented Dataset:

Skin Disease	**Samples**	**Source (TAG)**
acne	110	4
actinic-keratosis	110	4
basal-cell-carcinoma	110	4
chickenpox	110	3
cowpox	54	3
dermatitis	110	4
eczema	110	4
erythema-multiforme	110	4
granuloma	110	4
herpes	110	4
hidradenitis-suppurativa	101	4
lupus	110	4
measles	110	3
melanocytic-nevi	99	2

melanoma	110	2
molluscum	110	4
monkeypox	110	1,3
psoriasis	110	2
rosacea	110	4
tinea	110	2
vasculitis	110	4
warts	110	4
normal	110	3
TOTAL	2454	

Table 5.2 Unaugmented dataset

5.2.1 Augmentation

Image augmentation is a technique used in machine learning and computer vision to artificially increase the size of a dataset by generating new, modified versions of the original images. The goal of image augmentation is to improve the robustness and generalization ability of a model by exposing it to a wider range of variations and distortions in the training data.
Image augmentation can be achieved through a variety of techniques, including:
1. Flipping: The image is flipped horizontally or vertically.
2. Rotation: The image is rotated by a certain angle.
3. Scaling: The image is scaled up or down.
4. Translation: The image is shifted horizontally or vertically.
5. Shearing: The image is sheared along a certain axis.
6. Noise injection: Random noise is added to the image.
7. Color jittering: The color balance and contrast of the image is modified.
8. Elastic deformation: The image is distorted using elastic transformations.

By applying these transformations to the original images, image augmentation can create a diverse range of new images that are still semantically similar to the original images. This can help improve the accuracy and robustness of machine learning models, especially when dealing with complex and varied real-world data.

Image augmentation is commonly used in applications such as image classification, object detection, and semantic segmentation, where large amounts of training data are required for the model to generalize well to new data.

Augmented Dataset:

The following are the parameters set during Image Augmentation:
rotation_range=10
width_shift_range=0.1
height_shift_range=0.1
shear_range=0.1
zoom_range=0.1
horizontal_flip=True
vertical_flip=False

Skin Disease	**Test samples**	**Training samples**	**Validation samples**
acne	187	847	176
actinic-keratosis	187	847	176
basal-cell-carcinoma	187	847	176
chickenpox	187	847	176
cowpox	99	407	88
dermatitis	187	847	176
eczema	187	847	176
erythema-multiforme	187	847	176
granuloma	187	847	176
herpes	187	847	176
hidradenitis-suppurativa	176	770	165
lupus	187	847	176
measles	187	847	176
melanocytic-nevi	176	759	154

melanoma	187	847	176
molluscum	187	847	176
monkeypox	187	847	176
psoriasis	187	847	176
rosacea	187	847	176
tinea	187	847	176
vasculitis	187	847	176
warts	187	847	176
normal	187	847	176
TOTAL	4191	18876	3927

Table 5.3 Augmented dataset

CHAPTER 6

Background Studies

6.1 Artificial Neural Networks

6.1.1 Introduction

Artificial Neural Network, which is a type of machine learning algorithm inspired by the structure and function of biological neural networks found in the human brain. An ANN consists of interconnected nodes, called neurons, which are organized into layers.

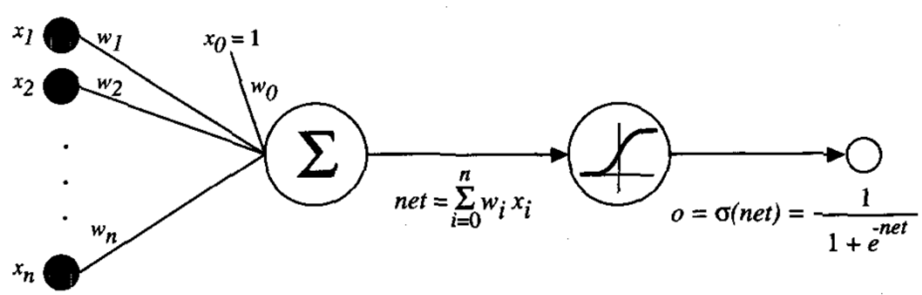


Figure 6.1 Perceptron

The first layer of an ANN is called the input layer, and it receives the data that is to be processed. The intermediate layers are called hidden layers, and they perform complex transformations on the input data. The final layer is called the output layer, and it produces the final result of the network's computation.

During the training process, an ANN adjusts the strength of the connections between neurons (synaptic weights) based on the input data and the desired output. This allows the network to learn and make predictions based on new data.

ANNs can be used for a variety of tasks, including image and speech recognition, natural language processing, and predictive analytics. They have shown great success in many applications, such as autonomous vehicles, fraud detection, and medical diagnosis.

6.1.2 Loss function in ANN

In Artificial Neural Networks (ANN), the loss function is a mathematical function that measures the difference between the predicted output and the actual output for a given set of input data.

The goal of training an ANN is to minimize the loss function by adjusting the weights of the network. The lower the value of the loss function, the better the network is at predicting the output for a given input.

Different types of problems require different loss functions. For example, for a binary classification problem, where the output can take only two values (e.g., yes or no), the commonly used loss function is Binary Cross-Entropy. For a multi-class classification problem, where the output can take multiple values (e.g., recognizing handwritten digits), the commonly used loss function is Categorical Cross-Entropy. For regression problems, where the output is a continuous value (e.g., predicting the price of a house), the commonly used loss function is Mean Squared Error.

During the training process, the weights of the network are updated in the direction that minimizes the loss function. This process is called backpropagation, where the error is propagated backwards from the output layer to the input layer, and the gradients of the loss function with respect to the weights are computed. The weights are then updated using an optimization algorithm, such as Gradient Descent, to minimize the loss function.

6.1.3 Regularization in ANN

Regularization is a technique used in ANN to prevent overfitting, where a network performs well on the training data but poorly on new data. It involves adding a penalty term to the loss function during training, which encourages the network to learn simpler patterns and reduces the chances of overfitting.

There are two common types of regularization: L1 and L2. L1 regularization adds a penalty term to the loss function proportional to the absolute value of the weights, while L2 adds a penalty term proportional to the square of the weights. Both help prevent overfitting by reducing the magnitude of the weights, but L1 produces sparse solutions, while L2 produces small, non-zero weights.

Other regularization techniques include dropout regularization, which randomly drops out some neurons during training, and early stopping, which stops training when the performance on a validation set deteriorates.

Regularization is crucial for improving generalization performance and preventing overfitting in ANN, leading to better performance on new data.

6.1.4 Advantages of ANN

Artificial Neural Networks (ANNs) have several advantages that make them a popular choice for many machine learning tasks:

1. Non-linearity: ANN captures nonlinear relationships between the input and output data.
2. Robustness: ANNs handle noisy data and missing values.
3. Real-world applications: ANNs have been successfully applied to a wide range of applications, including image and speech recognition etc.

6.2 Convolutional Neural Networks

6.2.1 Introduction

Convolutional Neural Networks (CNNs) are neural networks used for image recognition and computer vision tasks. They apply a series of convolutional filters to two-dimensional inputs, such as images or videos, for pattern recognition.

CNNs have multiple layers, including a convolutional layer, a pooling layer, and fully connected layers. The convolutional layer convolves filters over the input data to produce feature maps, highlighting specific patterns. The pooling layer reduces the dimensionality of the feature maps, while fully connected layers perform the final classification or regression task.

Overall, CNNs are a powerful tool for solving complex machine learning problems in computer vision.

6.2.2 Convolution Layer

Convolutional Neural Networks (CNNs) heavily rely on Convolutional layers as a fundamental element, which are commonly utilized in computer vision tasks such as object detection, image recognition, and segmentation.

The convolution process involves the movement of a filter across the input image, where the dot product of the filter weights and the corresponding region of the image is computed. This process is carried out for all possible locations of the filter, resulting in a feature map that highlights specific patterns within the input image. Each filter generates a single output feature map, with each element of the map corresponding to a specific region in the input image.

Additionally, Convolutional layers have other hyperparameters such as padding, filter size, and stride. The filter size determines the convolution's spatial range, while the stride specifies the amount by which the filter shifts over the input image. Padding is utilized to guarantee that the output feature map has the same dimensions as the input image.

Overall, Convolutional layers are critical for CNNs to learn vital features and patterns from images. By implementing these layers, CNNs can extract features automatically, eliminating the need for manual feature engineering. Therefore, CNNs are a potent tool for solving complicated machine learning challenges in computer vision.

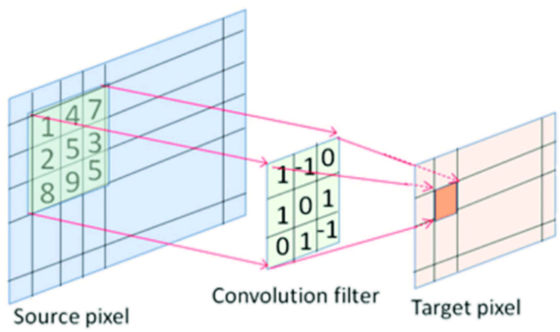


Figure 6.2 Working of a Convolution layer

6.2.3 Pooling Layer

Pooling layers are a significant component of Convolutional Neural Networks (CNNs) employed in computer vision tasks, such as image recognition. They are integrated after a convolutional layer to shrink the spatial dimensions of the feature maps and to extract robust features from them.

Max pooling is the most prevalent pooling method, which picks the maximum value within a rectangular neighborhood of the feature map. Other pooling techniques, such as average pooling, can also be utilized.

Pooling layers have hyperparameters, such as stride and pool size. The pool size defines the spatial area of the pooling operation, while the stride determines the amount by which the pooling window shifts over the input image.

Pooling layers offer several advantages, including reducing computation in subsequent layers and preventing overfitting. By implementing pooling layers, CNNs can derive robust features from high-dimensional input data, making them ideal for computer vision tasks, such as image recognition and object detection.

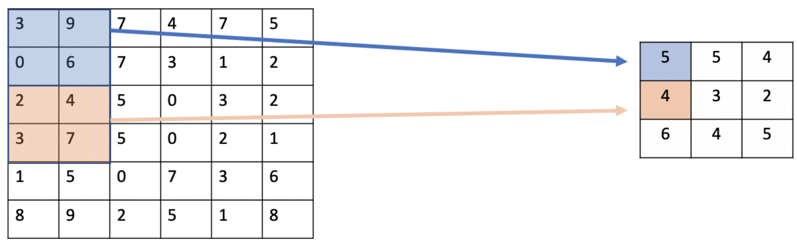


Figure 6.3 Average Pooling Visualization

6.2.4 Hyperparameters in CNN

Convolutional Neural Networks (CNNs) contain hyperparameters that require pre-setting by the user before initiating the training process. These parameters are not learned from data and include the number of filters, number of convolutional layers, activation function, regularization parameters, filter size, and stride. Other hyperparameters, such as learning rate, batch size, and the number of epochs, can also be adjusted.

Optimizing hyperparameters is a critical stage in constructing a CNN because it has a significant impact on model performance. The ideal combination of hyperparameters is reliant on the task at hand and the unique features of the data. Techniques like grid search, random search, and Bayesian optimization are available to efficiently explore the hyperparameter space and determine the best combination. In conclusion, selecting the optimal hyperparameters for a CNN is a time-consuming and iterative process but is essential for achieving optimal performance.

6.3 CNN Models

6.3.1 VGG

The VGG is a deep convolutional neural network design created to classify and recognize images. It was developed by researchers from the University of Oxford in 2014 and achieved superior performance on the ImageNet dataset. The VGG architecture includes a sequence of convolutional layers and max-pooling layers, and it concludes with three fully connected layers. The convolutional layers employ 3x3 filters with a stride of 1 and produce feature maps with the same spatial dimensions as the input. The max-pooling layers decrease the feature map's spatial dimensions by a factor of 2.

The VGG architecture's main innovation is the use of very deep networks with up to 19 layers, enabling the network to learn intricate features and patterns in the data, leading to improved performance[10]. Another important aspect is the use of small filters with a small stride, resulting in a better representation of the input data. As a result, the VGG network has become a well-known and significant architecture in the field of computer vision, inspiring many subsequent networks such as the ResNet and Inception architectures.

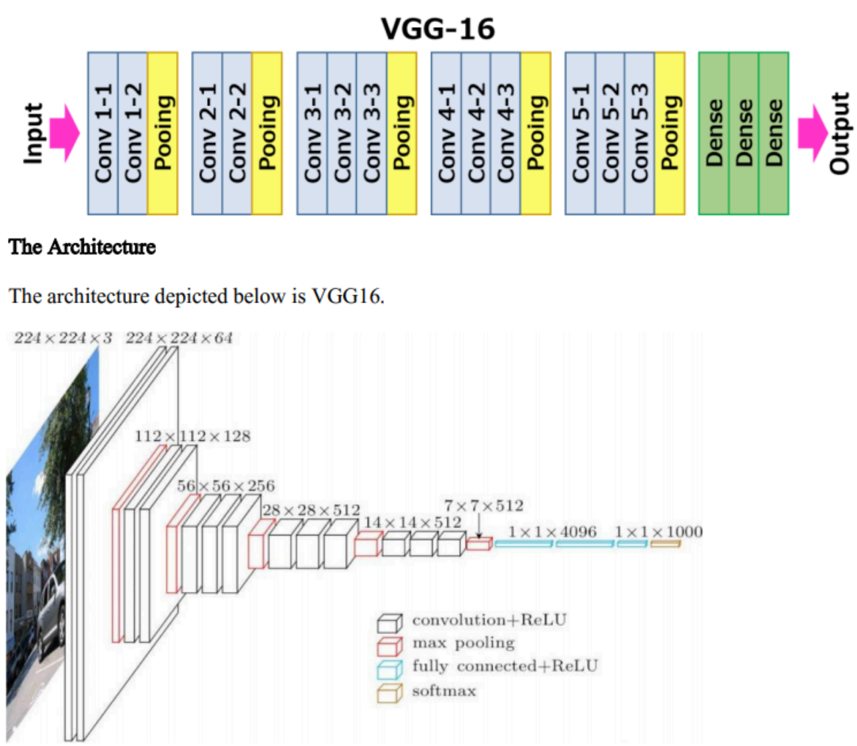


Figure 6.4 VGG-16 architecture

6.3.2 ResNet

The ResNet architecture consists of several important parts that work together to create a powerful deep neural network.[11]

1. Residual Connections: Residual connections are added between layers in the network to allow for better propagation of gradients during training. By adding a shortcut connection between layers, the network can learn to effectively propagate gradients even through very deep networks, addressing the problem of vanishing gradients.

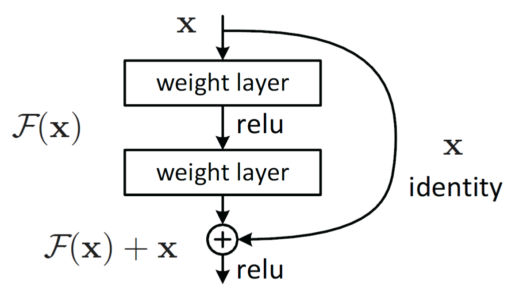


Figure 6.5 ResNet skip connection illustration

2. Bottleneck Layers: Bottleneck layers are used in ResNet to minimize the computational cost of the network. A bottleneck layer consists of a sequence of 1x1, 3x3, and 1x1 convolutions. The 1x1 convolutions are used to reduce the dimensionality of the input, while the 3x3 convolution is used to extract features. This allows the network to learn increasingly complex and abstract features while minimizing the computational cost of the network.

3. Skip Connections: Skip connections are used in ResNet to allow the network to learn features at different scales. By adding a shortcut connection between layers that skips over one or more layers in the network, the network can learn to detect features at different levels of abstraction. This can help to improve the accuracy of the network on tasks such as object detection and segmentation.

4. Deep Network Architecture: ResNet is known for its deep network architecture, with up to 152 layers. The use of a deep architecture allows the network to learn increasingly complex and abstract features, leading to improved performance on a variety of computer vision tasks.

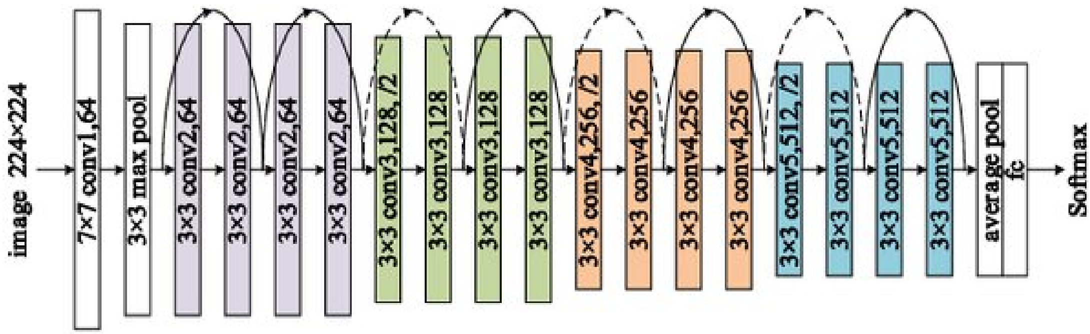


Figure 6.6 ResNet Architecture

Overall, the key components of the ResNet architecture work together to create a powerful deep neural network that can learn increasingly complex and abstract features while minimizing the computational cost of the network. The use of residual connections, bottleneck layers, skip connections, and a deep network architecture has led to significant improvements in performance on a variety of computer vision tasks.

6.3.3 MobileNet

MobileNet is a convolutional neural network architecture that was designed to be efficient and lightweight, making it well-suited for use in mobile and embedded devices where computational resources are limited.[12]

One of the key features of MobileNet is its use of depth wise separable convolutions. This type of convolutional operation separates the spatial and channel dimensions of the input, which reduces the computational cost of the operation. The depthwise separable convolution is made up of a depthwise convolution and a pointwise convolution, which applies a single filter to each input channel and then combines the output using a 1x1 convolution, respectively.

Another important aspect of MobileNet is its use of a width multiplier and a resolution multiplier. The width multiplier controls the number of channels in each layer, while the resolution multiplier controls the input image size. These parameters can be adjusted to optimize the network for a given set of computational resources, allowing for high accuracy while using minimal resources.

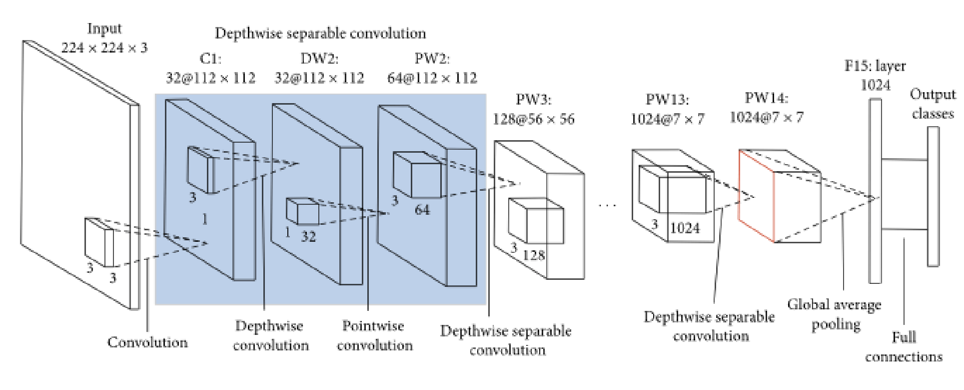


Figure 6.7 MobileNet architecture

MobileNet has achieved state-of-the-art performance on various computer vision tasks, such as image recognition, object detection, and segmentation. It is widely used in mobile and embedded applications, where computational resources are limited, but accurate and efficient neural networks are still required.

6.3.3.1 Depth Wise separable convolutions

Depthwise separable convolutions are a type of convolutional operation in which the spatial and channel dimensions of the input are separated, reducing the computational cost of the operation. This is achieved by applying a single filter to each input channel in a depthwise convolution, followed by a pointwise convolution that combines the output of the depthwise convolution using a 1x1 filter. The result is a more efficient convolution operation with fewer parameters, making it well-suited for use in mobile and embedded devices with limited computational resources.

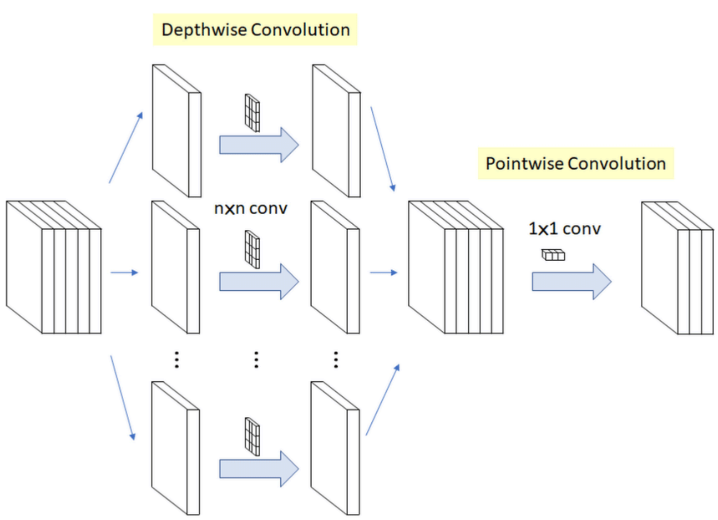


Figure 6.8 Depth wise Convolution

6.3.4 ShuffleNet

ShuffleNet is a convolutional neural network architecture that was introduced by researchers at Megvii in 2018. It is designed to be computationally efficient and well-suited for use in mobile and embedded devices.

ShuffleNet achieves its efficiency by using channel shuffling, which is a technique that enables information exchange between different groups of channels in the network. This is done using pointwise group convolutions, which convolve a subset of the input channels separately and then concatenate the results. The concatenated output is then shuffled, so that the different groups of channels can exchange information. This technique helps to reduce the number of parameters required in the network, making it more efficient.[13]

Another key feature of ShuffleNet is the use of depth wise separable convolutions, which are a form of convolutional operation that separates the spatial and channel dimensions of the input. This further reduces the computational cost of the operation, making the network more efficient.

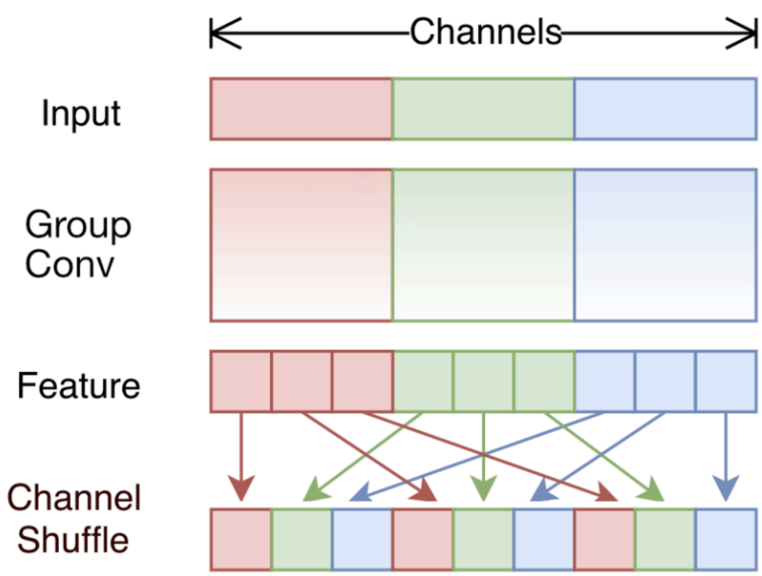


Figure 6.9 ShuffleNet shuffle channel mechanism

ShuffleNet has achieved state-of-the-art performance on various computer vision tasks, such as image classification and object detection, while using fewer parameters and less computational resources than other state-of-the-art models. It is widely used in mobile and embedded applications, where computational resources are limited, but accurate and efficient neural networks are still required.

6.3.5 Ensembling

Ensembling is a technique used in machine learning and deep learning to improve the performance and accuracy of a model by combining the predictions of multiple models. The basic idea behind ensembling is that by combining the predictions of multiple models that have been trained on the same dataset, the overall performance of the ensemble model can be improved.

There are various techniques used for ensembling, such as:

1. Bagging - Bagging involves training multiple models on different subsets of the training data, and then combining their predictions through a weighted average or voting scheme.

2. Boosting - Boosting involves training multiple models sequentially, with each model being trained on a modified version of the training data that puts more emphasis on the data points that were misclassified by the previous models.

3. Stacking - Stacking involves training multiple models and then using their predictions as inputs to another model, which then makes the final prediction.

Ensembling has been shown to be a powerful technique for improving the accuracy of machine learning models, and is widely used in applications such as image classification, natural language processing, and speech recognition. However, ensembling can also be computationally expensive and may require additional training data, so it is important to consider the trade-offs before using this technique.

6.4 Segmentation

6.4.1 Adaptive thresholding

Adaptive thresholding segmentation is a computer vision and image processing technique that is utilized to distinguish foreground objects from the background in an image. Unlike global thresholding, where a single threshold value is used for the entire image, adaptive thresholding varies the threshold value based on the local features of the image.

The standard thresholding approach can be inadequate when dealing with images that have varying lighting conditions, shadows, and noise. Adaptive thresholding addresses these challenges by using a different threshold value for each pixel based on the local features of the neighboring pixels.

The adaptive thresholding process involves dividing the image into smaller sub-regions or blocks, calculating a threshold value for each block based on the neighboring pixels' local features, and applying the threshold value to each pixel in the block to separate the foreground from the background.

6.4.2 Otsu's thresholding

Otsu thresholding is a method for image segmentation that automatically determines the optimal threshold value to separate the foreground and background in an image.

The Otsu thresholding method works by maximizing the variance between the foreground and background pixel intensities. The optimal threshold value is determined by iterating over all possible threshold values and computing a measure of the separability between the foreground and background pixel intensities at each threshold. The measure of separability is typically the variance between the two groups of pixels.

Once the measure of separability has been computed for all possible threshold values, the threshold that maximizes the separability measure is chosen as the optimal threshold value. This threshold value is then used to segment the image into foreground and background regions.

Otsu thresholding is a powerful and widely used method for image segmentation, especially in situations where the foreground and background pixel intensities have bimodal distributions. The method is relatively simple and computationally efficient, making it well-suited for real-time applications such as video processing and computer vision systems.

CHAPTER 7

Methodology

7.1 Image Augmentation

For augmentation, we took the selected images and split them into training, testing and validation set in the ratio of 70 : 15 : 15.
The libraries in python that we used to perform segmentation are os, cv2, shutil, ImageDataGenerator.

- os - used to traverse through the image folders
- cv2 - used to read/write the images
- shutil - move folders
- ImageDataGenerator - used to define the parameters of image augmentation

The following are the parameters set during Image Augmentation:
rotation_range=10
width_shift_range=0.1
height_shift_range=0.1
shear_range=0.1
zoom_range=0.1
horizontal_flip=True
vertical_flip=False

7.2 Image Segmentation

The Image segmentation turned out to be a complex problem in our scenario because there were various different types of images present in our dataset.

7.2.1 Challenges faced

Types of images found:

1. Full body images: These were the type of images that showed more than 50% of the human body
2. Body part images: These were the type of images that showed a certain part of the human body. Ex: face, arm, leg etc.
3. Skin only images: These were the type of images that only had skin in the complete frame

For each type of image, a different segmentation approach needed to be implemented.

- Full body image required OTSU's thresholding for segmentation because it contained bimodal type of distribution in the ROI and background
- Body part image required Adaptive thresholding for segmentation because human body parts can be of any shape and size in any given image
- Skin only image did not require any segmentation because, we already had the required ROI in it

7.2.2 Solution

To identify which type of segmentation is required for the given image, we trained a ML model to first identify which type of image is given.

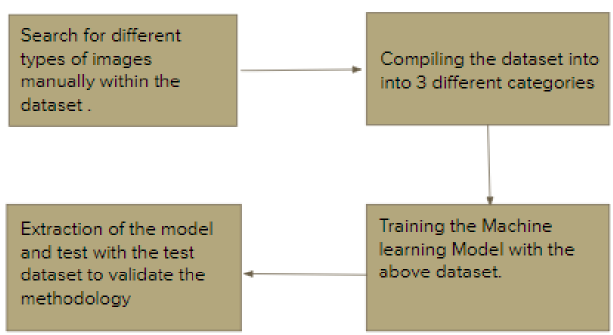


Figure 7.1 Proposed flow diagram of image-type identification model construction

7.2.2.1 Dataset description

Custom Dataset Construction:

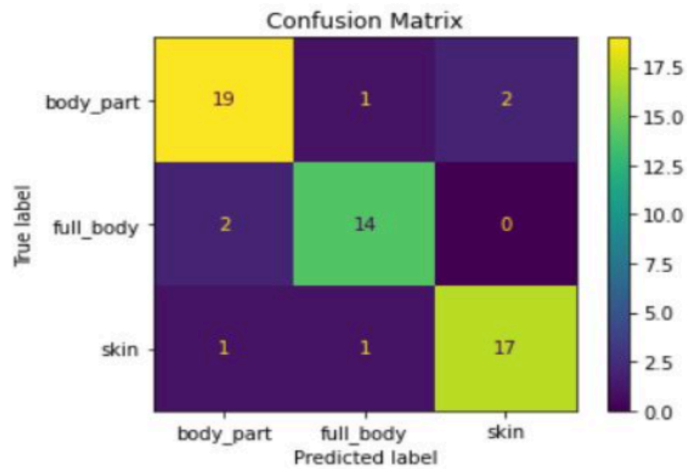
Image type	Test	Train	Validation
Full body	16	68	22
Body Part	22	82	31
Skin only	19	76	25
total	57	226	78

Table 7.1 Image-type dataset description

7.2.2.2 Machine Learning Model

After constructing the custom dataset, we trained a VGG19 model with the help of transfer learning over imagenet dataset for 100 epochs to obtain the classifier ML model with loss function as categorical_crossentropy and metrics as accuracy.

This is the confusion matrix obtained at the end during testing phase of the ML model:



Classification Report

	precision	recall	f1-score	support
0	0.86	0.86	0.86	22
1	0.88	0.88	0.88	16
2	0.89	0.89	0.89	19
accuracy			0.88	57
macro avg	0.88	0.88	0.88	57
weighted avg	0.88	0.88	0.88	57

Figure 7.2 Confusion Matrix for image-type identification model

7.2.2.3 Segmentation procedure

Now that we had a way to identify which type of image is given, we used the labels generated for the given image from the ML model to use the required segmentation technique for it.

7.2.2.4 Conclusion

The procedure of segmentation can be summarized as follows:

1. Given image is input into the ML model to identify which type of image is given
2. The segmentation technique is the applied based on the label generated by the ML model
3. After segmentation is applied on the given image, the new segmented image is the saved/forwarded to the skin disease classifier.

7.3 Model Training and Selection

7.3.1 VGG19

We used the VGG19 based model from keras.applications.vgg19 with weights as ‘imagenet’ for transfer learning in this case.

Activation function at the end was set as softmax, we used Adam optimiser with learning rate of ‘0.001’ and decay of 0.01/20.

The model was compiled with the help of the above optimiser, ‘categorical_crossentropy’ as loss function and ‘accuracy’ as metrics.

While training, the number of epochs were set to 40.

Testing Results:

Model	Accuracy	Precision	Recall
VGG19	91.2%	90.1%	94.2%

Table 7.2 VGG19 test results

7.3.2 MobileNet

We used the MobileNetV2 model from keras.applications.mobilenet_v2 with weights set as that of ‘imagenet’ for transfer learning. The optimiser we chose was Adam with a learning rate of 0.0001. We also unfroze the last 20 layers of the mobilenetv2 base model to fine tune the moilenetv2 model.

We compiled the model with the above given optimiser, categorical_crossentropy as loss function and accuracy as metrics.

The input size for the model is (224,224,3) and output is the probability distribution over the given 23 classes of diseases (22 skin diseases and normal skin)

We set the number of epochs as 15 with a batch size of 32.

Testing results:

Model	Accuracy	Precision	Recall
MobileNetV2	91.25%	89.1%	95.2%

Table 7.3 MobileNet test results

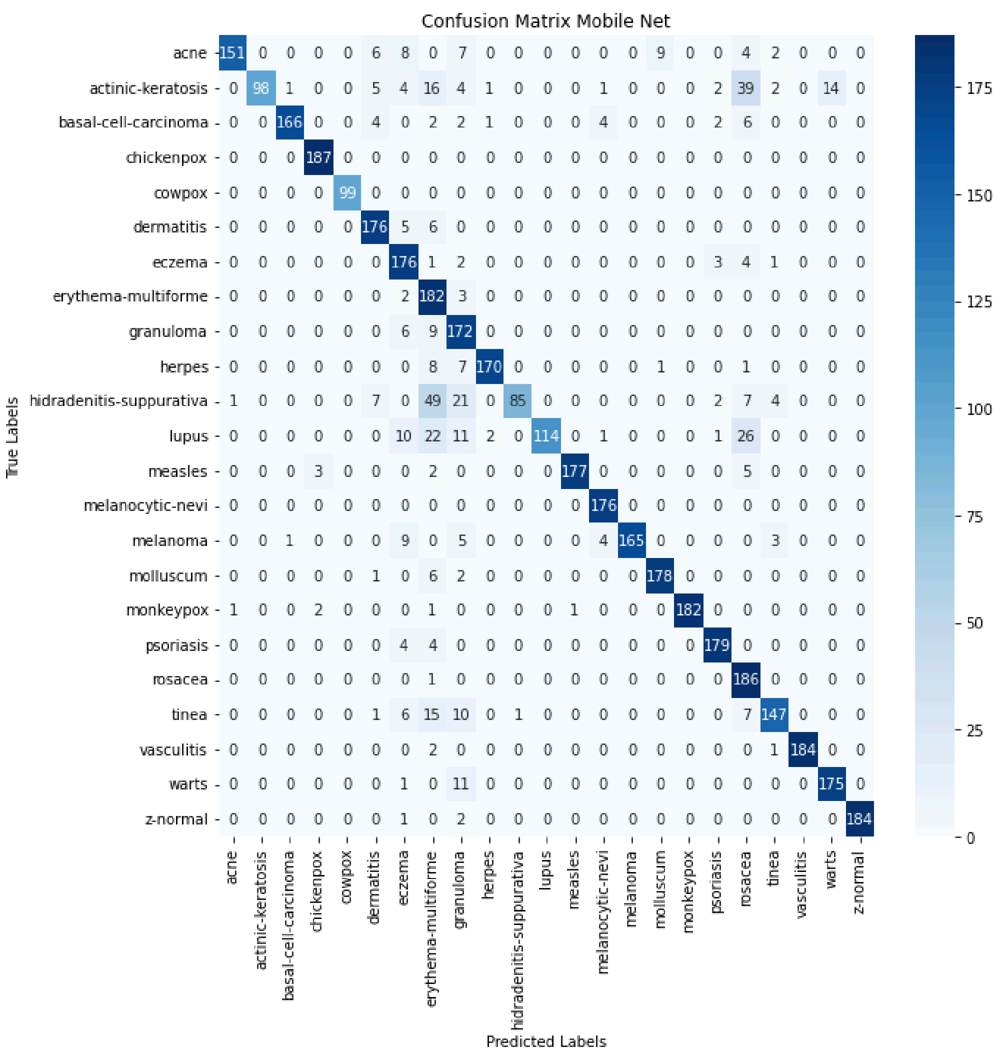


Figure 7.3 Confusion Matrix for MobileNetV2

7.3.3 ShuffleNet

For ShuffleNet, we implemented it from scratch. The implementation consists of 3 stages and an output layer.

Stage 1 consists of :

1. Convolution layer with 24 filters and (3,3) kernel_size with strides = (2,2)
2. BatchNormalization
3. Activation function (relu)
4. MaxPooling2D layer

Stage 2 consists of 3 shuffle units, each shuffle unit consists 2 branches:

- Branch 1 consists of:
    1. DepthwiseConv2D layer with (3,3) kernel_size, stride = (1,1)
    2. BatchNormalization
    3. Conv2D layer
    4. BatchNormalization
    5. Relu Activation function
- Branch 2 consists of:
    1. Conv2D layer
    2. BatchNormalization
    3. Relu activation function
    4. DepthwiseConv2D layer with (3,3) kernel_size, stride = (1,1)
    5. BatchNormalization
    6. Conv2D layer
    7. BatchNormalization
    8. Relu Activation function

Stage 3 consists of:

1. Conv2D layer with 384 filters, kernel_size = (1,1) and stride = (1,1)
2. BatchNormalization
3. Relu Activation function
4. GlobalAveragePooling2D

Output layer consists of 1 Dense layer with units = 23 and softmax activation function

In shuffle units, we are using a shuffle_channels function to shuffle channels for the implementation.

We then compile the model with ‘Adam’ optimizer with default learning rate, ‘categorical_crossentropy’ as the loss function and ‘accuracy’ as metrics.

The model is then trained for 20 epochs with a batch size of 32.

Testing results:

Model	Accuracy	Precision	Recall
ShuffleNet	78.14%	76%	82%

Table 7.4 ShuffleNet test results

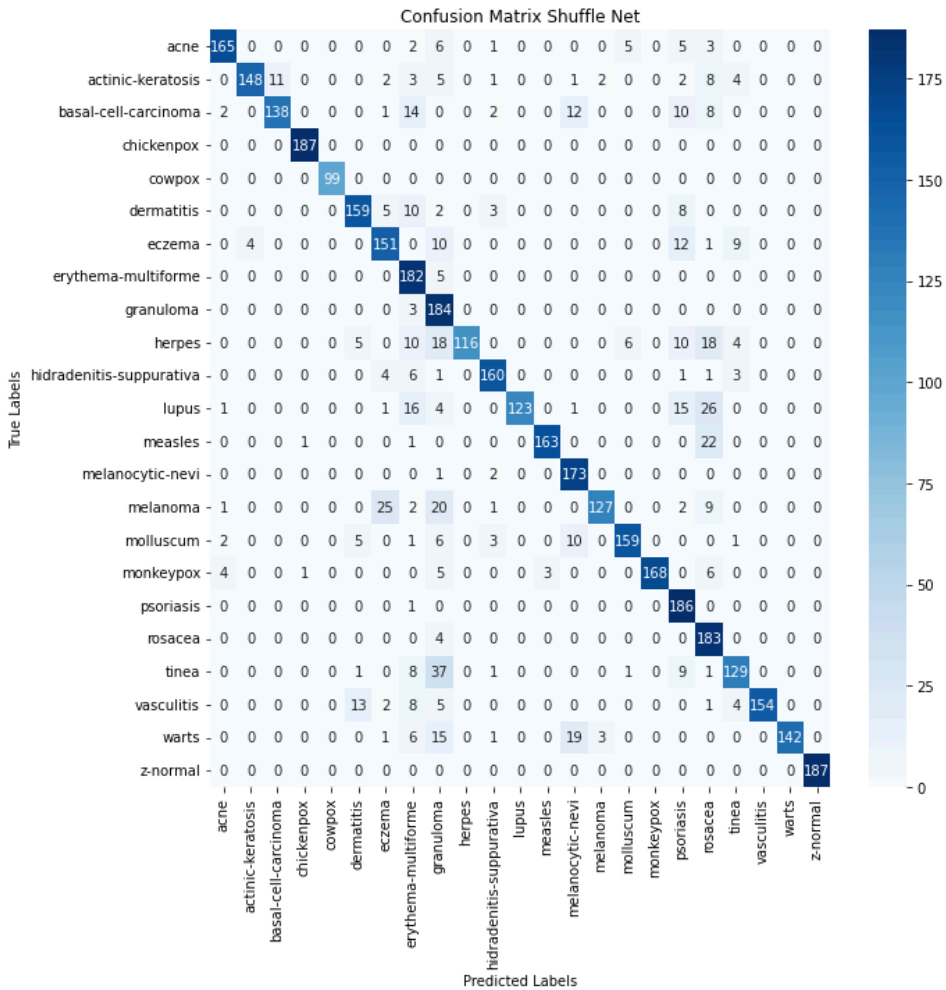


Figure 7.4 Confusion Matrix for ShuffleNet

Model	Accuracy
Ensemble (average) of MobileNetV2 and ShuffleNet	93.2%

Table 7.5 Ensemble Accuracy

7.3.4 Ensembling

For the ensemble, we are using the ShuffleNet and MobileNetV2 model.

The ensembling method we used was average ensembling, where we take the outputs of both the models and average them to find out the most probable answer.

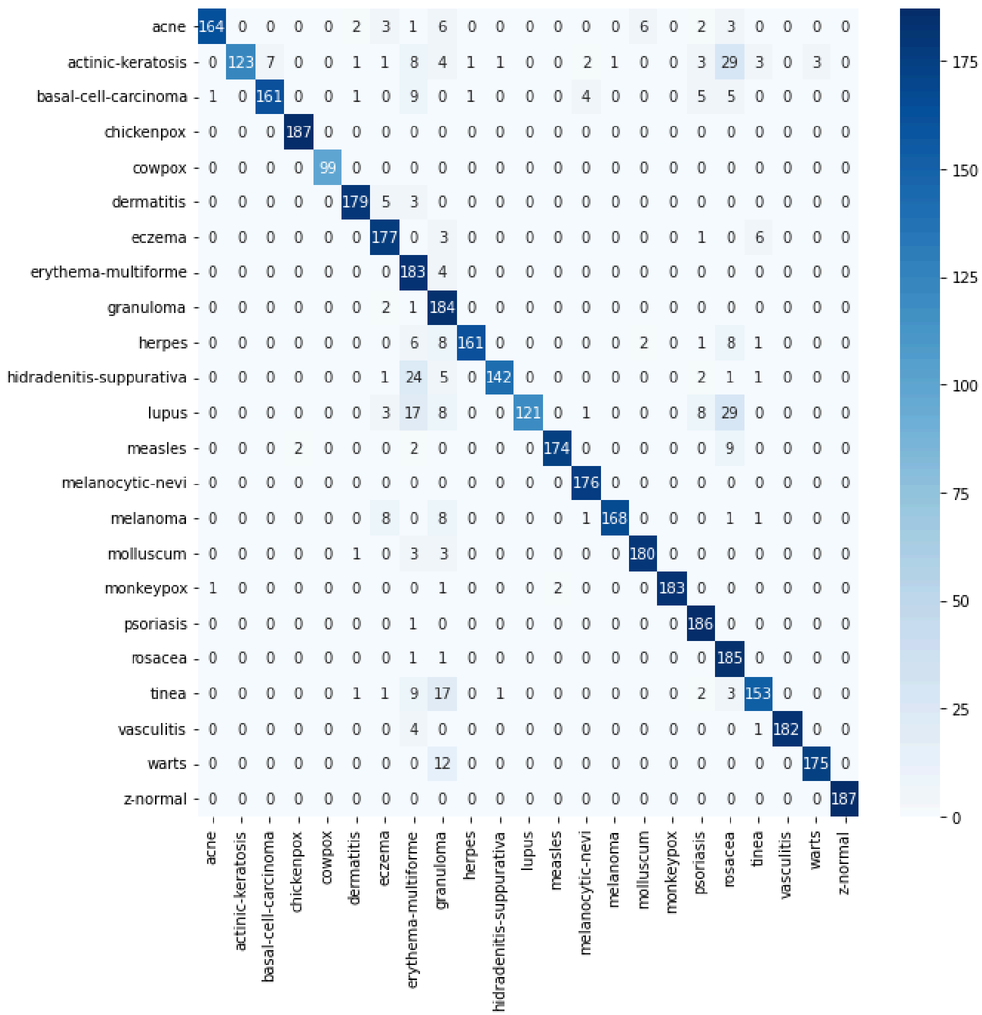


Figure 7.5 Confusion Matrix for Ensemble of MobileNetV2 and ShuffleNet

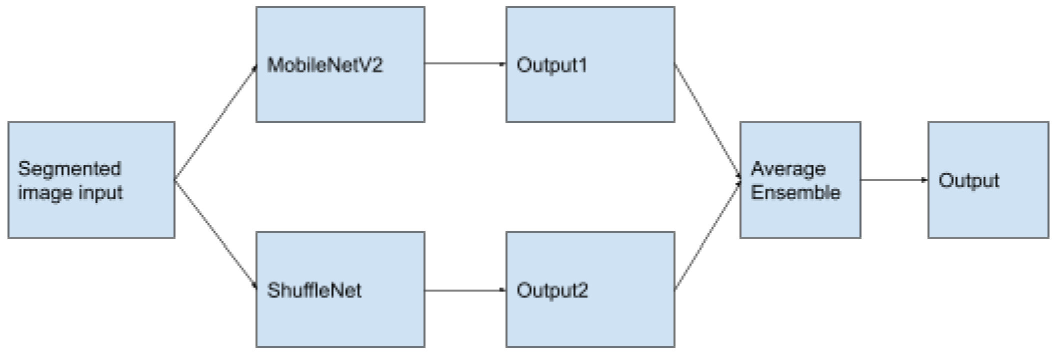


Figure 7.6 Ensemble Architecture

7.3.5 Model Selection

From the above results we select the Ensembling of MobileNetV2 and Shufflenet as our Machine Learning model for the project. We are not choosing VGG19 because Ensemble is giving better results than VGG19 and the number of parameters of Ensemble is also less than VGG19. Hence, Ensemble is faster and more accurate than VGG19

7.4 Solution Implementation

The following is the flow of the solution:

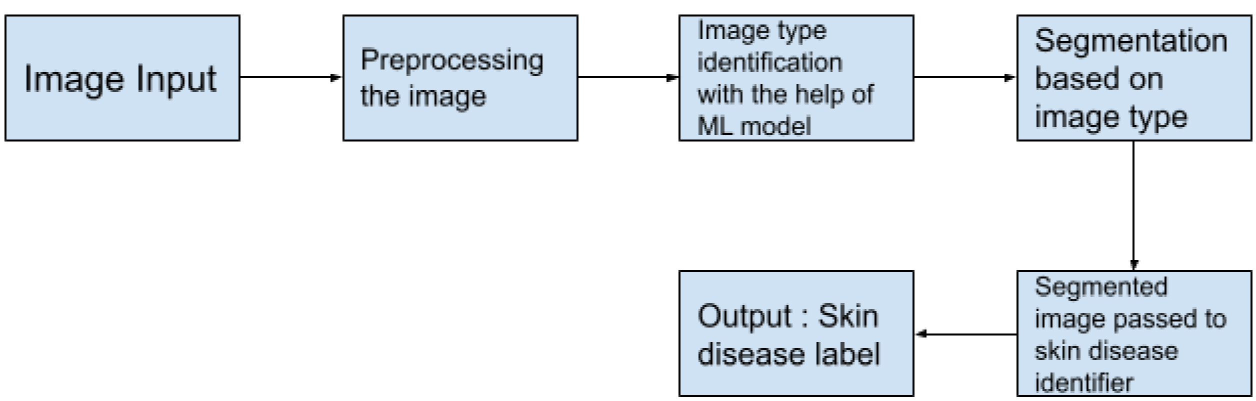


Figure 7.7 Skin disease identifier flow chart

7.4.1 Image Input

User is required to give the image input. The image can be of any size preferably (224,224,3).

7.4.2 Image Preprocessing

The given image is then resized to the required size of (224,224,3).The pixel values are then normalized for better results during the preprocessing phase of the system.

7.4.3 Image type identification

This image is then passed onto the image type identification model to identify which type of image we are handling with so that we can use the appropriate segmentation approach on the image.

7.4.4 Segmentation

With respect to the assigned label to the image, segmentation is done and then the segmented image is then passed to the skin disease classifier ML ensemble model.

7.4.5 Skin disease classification using the trained ML model

The ensemble model takes the segmented image and generates the probability distribution of the likelihood of it being a certain disease taken in this project.

7.4.6 Output

From the generated probability distribution for the image, the maximum value is chosen and the corresponding skin disease label is then identified.

CHAPTER 8

CONCLUSION

8.1 Summary

To summarize, our paper reviewed the existing papers on skin disease classification systems, identified their shortcomings and suggested a new approach to identify skin diseases using segmentation and CNN.

We first combined certain datasets to construct our own dataset and then we ensured balanced distribution of skin disease samples within the dataset. We augmented the dataset using ImageDataGenerator and gave a new segmentation methodology for segmenting the images within the dataset.

After Segmentation we studied various machine learning models and chose the most appropriate approach to solve our problem. We chose average ensembling of MobileNetV2 and ShuffleNet which gave testing accuracy of 93.2% over the respective models chosen separately in MobileNet gave 91.25% testing accuracy and ShuffleNet gave 78.14% testing accuracy and VGG19 gave 91.2% testing accuracy.

At the end we suggested the complete flow of the solution to identify skin diseases.

REFERENCES

[1] Md Manjurul Ahsan, Muhammad Ramiz Uddin, Mithila Farjana, Ahmed Nazmus Sakib, Khondhaker Al Momin, Shahana Akter Luna, “Image Data collection and implementation of deep learning-based model in detecting Monkeypox disease using modified VGG16” (June 2022)

[2] Shams Nafisa Ali , Md. Tazuddin Ahmed1 , Joydip Paul1 , Tasnim Jahan1 , S. M. Sakeef Sani1 , Nawsabah Noor , Taufiq Hasan1,”Monkeypox Skin Lesion Detection Using Deep Learning Models: A Feasibility Study”, Department of Biomedical Engineering, Bangladesh University of Engineering and Technology Popular Medical College, Dhaka, Bangladesh

[3] Shaden Abdulaziz AlDera, Mohamed Tahar Ben Othman, “A Model for Classification and Diagnosis of Skin Disease using Machine Learning and Image Processing Techniques” in International Journal of Advanced Computer Science and Applications(IJACSA), Volume 13 Issue 5, 2022.

[4] K. Roy, S. S. Chaudhuri, S. Ghosh, S. K. Dutta, P. Chakraborty and R. Sarkar, "Skin Disease detection based on different Segmentation Techniques," 2019 International Conference on Opto-Electronics and Applied Optics (Optronix), Kolkata, India, 2019, pp. 1-5, doi: 10.1109/OPTRONIX.2019.8862403.

[5] T. Swapna , D.A. Vineela , M. Navyasree , N. Sushmtha ,P. Bhavana, “Detection and Classification of Skin diseases using Deep Learning ” in The International journal of analytical and experimental modal analysis Volume XIII, Issue VIII, August/2021 ISSN NO:0886-9367

[6] “Monkeypox Skin Lesion Dataset”,
https://www.kaggle.com/datasets/nafin59/monkeypox-skin-lesion-dataset

[7] “Skin diseases image dataset”,
https://www.kaggle.com/datasets/ismailpromus/skin-diseases-image-dataset

[8] “A Web-scraped Skin Image Database of Monkeypox, Chickenpox, Smallpox, Cowpox, and Measles”, https://www.biorxiv.org/content/10.1101/2022.08.01.502199v2

[9] “Dermnet”, https://www.kaggle.com/datasets/shubhamgoel27/dermnet

[10] Karen Simonyan, Andrew Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition” in arXiv:1409.1556v6 [cs.CV]

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, “Deep Residual Learning for Image Recognition”, arXiv:1512.03385v1 [cs.CV]

[12] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”, arXiv:1704.04861v1 [cs.CV]

[13] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, Jian Sun, “ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices”, arXiv:1707.01083v2 [cs.CV]