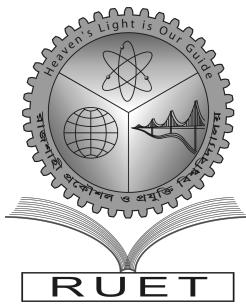


Heaven's Light is Our Guide



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Rajshahi University of Engineering & Technology, Bangladesh

Classification of skin disease using deep learning neural networks

Author

Sumiaya Akter & Mst. Fatema Khatun

Roll No. 1803018, 1803033

Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology

Supervised by

Prof. Dr. Boshir Ahmed

Professor

Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology

ACKNOWLEDGEMENT

First and foremost, we want to convey our gratitude to the Almighty Allah for giving his showers of blessing and provides the opportunity and enthusiasm throughout our research work to complete it successfully.

We would like to express our deep and sincere gratitude and respect to our supervisor Dr. Boshir Ahmed, Professor. Department of Computer Science & Engineering, Rajshahi University of Engineering & Technology, Rajshahi. He has given us the opportunity to do our research and gives his invaluable guidelines, advice and necessary documents to complete the work. His dynamism and sincerity inspire us greatly in all possible way. Whenever we got struck or faced any problem, he was always there for us at any time. Without his continuous support this work has not been materialized in the final form that it is now at the present. We are extremely grateful for getting the opportunity to work with him under his guidance.

We are also grateful to all the respective teachers of Computer Science & Engineering, Rajshahi University of Engineering & Technology, Rajshahi for giving their valuable suggestions, advice and inspirations from time to time.

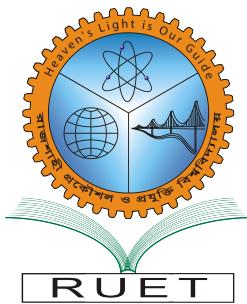
Finally, we want to give thanks to our parents, friends and well-wishers for their constant support and inspirations throughout this journey.

Date:

Sumiaya Akter & Mst. Fatema Khatun

RUET, Rajshahi

Heaven's Light is Our Guide



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Rajshahi University of Engineering & Technology, Bangladesh

CERTIFICATE

*This is to certify that this thesis report entitled “**Classification of skin disease using deep learning neural networks** ” submitted by **Sumiaya Akter & Mst. Fatema Khatun, Roll:1803018, 1803033** in partial fulfillment of the requirement for the award of the degree of Bachelor of Science in Department of Computer Science & Engineering of Rajshahi University of Engineering & Technology, Bangladesh is a record of the candidate own work carried out by him under my supervision. This thesis has not been submitted for the award of any other degree.*

Supervisor

External Examiner

Prof. Dr. Boshir Ahmed

Professor

Department of Computer Science &
Engineering
Rajshahi University of Engineering &
Technology
Rajshahi-6204

External Examiner’s Name

External Examiner’s Designation

Department of Computer Science &
Engineering
Rajshahi University of Engineering &
Technology
Rajshahi-6204

ABSTRACT

Skin disease can present with a wide range of symptoms that overlap, making diagnosis and therapy very challenging. Through the use of large datasets and complex patterns found in medical imaging, deep learning algorithms offer practical solutions for diagnosing automation. We describe a novel approach to skin disease classification using a hybrid deep learning model that incorporates Long Short-Term Memory (LSTM) networks with Convolutional Neural Networks (CNNs). The MobileNet V2 architecture is the feature extractor; it efficiently extracts relevant information from skin lesion photographs while lowering computer complexity. The gathered characteristics are then put into LSTM layers to model temporal dependencies and find sequential patterns in the data. This hybrid architecture allows the network to learn both spatial and temporal data effectively, improving classification accuracy. Key preprocessing techniques include normalization, where image data is scaled to a consistent range (such as 0 to 1) and resized to a standard dimension (like 224x224 pixels), which is crucial for consistent input across models. The images' performance has been enhanced by normalization and the use of different data augmentation techniques has increased the dataset's diversity. We evaluate our suggested model using a publicly available dataset that includes images of a range of skin conditions, including melanoma, basal cell carcinoma, and benign lesions. As measure metrics, we used accuracy, loss, precision, recall, and F1 score to assess the models. The accuracy score of the model was 88%. Deep learning models that are biased may result from datasets that have an uneven distribution of classes associated with skin disorders. This would diminish the therapeutic potential of the models and make them harder to interpret and generalize. Self-learning features, past experience data, and randomization components can help the model reduce training effort and increase pattern exploration efficiency, while the residual connection bottleneck allows MobileNet V2 with LSTM to achieve better accuracy with less computational overhead.

CONTENTS

ACKNOWLEDGEMENT

CERTIFICATE

ABSTRACT

CHAPTER 1

Introduction	1
1.1 Introduction	1
1.2 Problem Statement	2
1.3 Research Challenges	3
1.4 Objectives	4
1.5 Outlines	4
1.6 Conclusion	5

CHAPTER 2

Skin Disease	6
2.1 Introduction	6
2.2 History of skin disease	6
2.3 Epidemiology of Skin Disease	7
2.4 Several kinds of Skin Disease	7
2.4.1 Melanocytic nevi	7
2.4.2 Benign keratosis	9
2.4.3 Dermatofibroma	10
2.4.4 Vascular lesions	11
2.4.5 Actinic keratoses	11
2.4.6 Intraepithelial carcinoma	12
2.4.7 Basal cell carcinoma	12
2.4.8 Melanoma	13

2.5 Conclusion	15
 CHAPTER 3	
Background Study and Literature Review	16
3.1 Introduction	16
3.2 CNN Architecture	16
3.2.1 Convolutional Layer	17
3.2.2 Pooling Layer	17
3.2.3 Fully Connected Layer & Activation funtions	18
3.3 Deep CNN Architecture	18
3.4 Overall deep CNN model architecture	19
3.5 Deep Residual Network	19
3.6 Residual Blocks	20
3.7 Fully Convolutional Residual Network	21
3.8 Multi-Scale CNN	22
3.9 Inception v3	24
3.10 Mobilenet	25
3.11 Support Vector Machine	26
3.12 KNN	27
3.13 Naïve Bayes Classifier	28
3.14 Back Propagation Neural Network Algorithm	28
3.15 Literature Review	30
 CHAPTER 4	
Methodology	32
4.1 Introduction	32
4.2 Methodology	32
4.3 Overview of Proposed Methodology	33
4.4 Dataset Descriptions	34
4.5 Dataset Preprocessing	35
4.5.1 Normalization	36
4.5.2 Image resizing	36
4.5.3 Data Augmentation	36

LIST OF TABLES

4.1	HAM10000 Dataset	35
4.2	Data Augmentation Techniques	37
4.3	Training Subset	38
4.4	Validation Subset	38
4.5	Validation Subset	45
5.1	Below some epoch and their values	49
5.2	After Fine Tuning some epoch and their values	50

4.4	Mobilenet V2 Architecture	40
4.5	Mobilenet V2	41
4.6	LSTM Architecture	42
4.7	Mobilenet V2 with LSTM	43
5.1	Training and validation Accuracy 1	49
5.2	Training and validation loss 1	50
5.3	Training and validation Accuracy 2	51
5.4	Training and validation Loss 2	51
5.5	confusion matrix 1	52
5.6	confusion matrix 2	52
5.7	classification report	53

Chapter 1

Introduction

1.1 Introduction

Skin disease is one of the most common ailments affecting people worldwide. Melanoma, intraepithelial carcinoma, squamous cell carcinoma (SCC), and basal cell carcinoma (BCC) are examples of skin cancers.

These abnormalities are important in the diagnosis of many inside ailments, even though most skin diseases originate in the skin's layers. The theory that a person's skin indicates their inside health has some merit. The skin often shows symptoms of an underlying ailment before any other organ since it is easily accessible and visible. Skin anomalies frequently point to metabolic, neoplastic, and glandular diseases.

Like other tissues, skin can experience a wide range of pathological changes, including inflammatory, benign, hereditary, and more[1].

Seven skin illnesses were included in the dataset utilized in this study: melanocytic nevi, vascular lesions, actinic keratoses, intraepithelial carcinoma, basal cell carcinoma, dermatofibroma, and melanoma. Over 10,000 dermatoscopic pictures are included in this dataset. After applying a random function, the data is divided into two categories: training data (7224) and validation data (1255)[2].

1.2 Problem Statement

The suggested problem is to combine the Long Short-Term Memory (LSTM) and MobileNet V2 neural network architectures to create a reliable and automated method for classifying skin disorders.

Using the MobileNetV2 architecture, which is specifically designed for mobile and embedded vision applications, the goal of the skin illness classification problem is to create a reliable and automated system for correctly classifying different skin disorders. MobileNetV2 is an excellent choice for mobile device deployment because to its lightweight design, which minimizes computing resources and allows for effective feature extraction through the use of depth-wise separable convolutions and efficient inverted residuals.

The system aims to improve dermatological diagnosis by quickly and accurately classifying skin illnesses using photographs taken with smartphones or other portable devices, utilizing MobileNetV2's remarkable accuracy and efficiency. This solution empowers users to conduct preliminary evaluations and seek necessary medical assistance swiftly, thereby meeting the growing demand for efficient and easily accessible healthcare solutions, especially in areas with limited access to specialized medical facilities.

In the context of skin disease classification, one important element that stands out is LSTM (Long Short-Term Memory), which is unique in that it can identify long-term dependencies in sequential data. The model gets the capacity to recognize temporal patterns present in relevant data sequences, like pictures showing skin problems or patient medical histories, by including LSTM into the classification pipeline. As a result, the model may identify subtle relationships over time, leading to a more precise and knowledgeable classification of diseases.

The performance of the model is improved by utilizing LSTM's capacity to retain and use information over long sequences, which enables more thorough investigation and diagnosis of skin conditions. In the end, this integration increases the effectiveness of the automated skin disease classification system by addressing the dynamic nature of skin disorders and the significance of taking temporal information into account for accurate categorization.

of the suggested model is also provided.

Chapter 5

The statistical findings for the model are presented in this chapter. For better understanding, we have shown a performance analysis. A simple confusion matrix is also provided where percentage of accuracy and loss is mentioned.

Chapter 6

In this chapter, we have given a overview of our work and also made an effort to provide a new view so that the future work will be done more effectively.

1.6 Conclusion

In conclusion, even in contexts with limited resources, the combination of MobileNet V2 and LSTM has shown impressive efficiency for the categorization and detection of skin diseases. By taking use of temporal relationships in sequential data, the LSTM module ensures that the model can retain context from prior timestamp data, which improves prediction accuracy. It's important to note, nevertheless, that even with its general effectiveness, the model's precision dropped noticeably to less than 80% when evaluated on a set of images taken in low light. This draws attention to a possible shortcoming of the model, especially in situations where image quality is impaired.

Chapter 2

Skin Disease

2.1 Introduction

Skin diseases encompass a broad spectrum of skin-related issues, ranging from mild irritations to severe and debilitating conditions. These disorders may be caused by a variety of causes, including genetics, infections, allergies, autoimmune reactions, environmental triggers, and lifestyle decisions. Researchers are crucial to our understanding of and ability to cure skin diseases. Researchers improve our understanding of and capacity to treat skin disorders in a variety of ways. Their efforts contribute to the development of new treatments, improved diagnostic methods, preventative measures, and ultimately improved outcomes for those suffering from these conditions.

2.2 History of skin disease

A surge in interest in the medical sciences, particularly skin research, was sparked by the Renaissance, which took place in the 15th and 16th centuries and saw a tremendous expansion of knowledge and a reappraisal of Greek logic. Thus, concepts of skin architecture and topical therapy expand, and the first categories of dermatological illnesses emerge. Daniel Turner (1667–1741), who was regarded as the founder of British dermatology until the early 1900s, sparked interest in the field more than a century after Mercuriale with his work *De Morbis Cutaneis*. Published in 1714, the treatise *De Morbis Cutaneis* had more than 100 clinical instances of dermatological disorders along with their corresponding cures[3]. It was subsequently translated into many languages and published in multiple editions.

The think about of skin infections has been connected to common pharmaceutical for centuries. As it were within the 18th century, driven by the progression of science and scientific categorization within the areas of information, did the primary writings and works committed particularly to the think about of skin infections develop[3].

Europe saw a surge in dermatology research during the 18th and 19th centuries, with three prominent medical and research hubs emerging: the UK, France, and Austria[4]. The science of dermatology owes its roots to dermatologists from the British, French, and Austrian schools, whose discoveries, theories, and knowledge continue to influence the field today[3].

The 20th-century scientific and technical revolution brought new surgical and cosmetic techniques as well as new therapeutic resources, which completely changed the field of dermatological care[3].

2.3 Epidemiology of Skin Disease

Based on data from 2016 to 2020, the yearly incidence of cutaneous melanoma is 21.0 instances per 100,000 men and women, with a death rate of 2.1 per 100,000[5]. Data from 2017 to 2019 indicate that 2.2% of men and women may receive a skin melanoma diagnosis at some point in their lives.

The most recent research states that between 2008 and 2018, there was an annual increase of 53% in the number of new cases of melanoma diagnosed. The proportion of newly diagnosed cases of non-melanoma cancer rose sharply to 77% between 1994 and 2014[6]. With 3000 deaths each year from non-melanoma skin cancer, basal cell carcinoma is the most common kind[7].

2.4 Several kinds of Skin Disease

2.4.1 Melanocytic nevi

All skin types exhibit melanocytic nevi, also known as moles. Melanocytic nevi are often classified as either acquired or congenital. A patient typically develops more nevi during childhood and adolescence, with the third decade of life marking the end of fresh mole acquisition. Due to the wide variation in the morphology and presentation of melanocytic nevi, distinct descriptive

- Varied quantity, from one growth to several growths.
- Often referred to as flesh moles or dermatosis papulosa nigra, these tiny growths are grouped around the eyes or other parts of the face and are typically seen on people with black or brown skin.
- Varied in hue, from pale tan to dark brown to black.
- Itching.

2.4.3 Dermatofibroma

Common skin growths are called dermatofibromas or fibrous histiocytomas. These are firm or hard to the touch papules, which are little, smooth, solid lumps, or nodules, which are larger, smooth, solid bumps. They may have a white scar-like center and might be skin-colored, pink, brown, reddish, or purple in color[11]. Although they are rarely unpleasant or irritating, dermatofibromas can occasionally be sensitive to the touch. They last a lifetime on average, however after a few years, they may grow softer and flatter. If there are a lot of dermatofibromas, there can be an underlying illness like lupus or a compromised immune system from leukemia or HIV.



Figure 2.3: Dermatofibroma[2]

Symptoms

Women's arms and legs are the most common places to see dermatofibromas. These are nodules or papules that can be any shade of brown in darker skin tones or tan, pink, red, or brown in lighter skin tones [11]. There can be a white center to them. When touched, they may occasionally feel sensitive, but mostly they feel rather hard or even firm. A "dimple sign," in which the

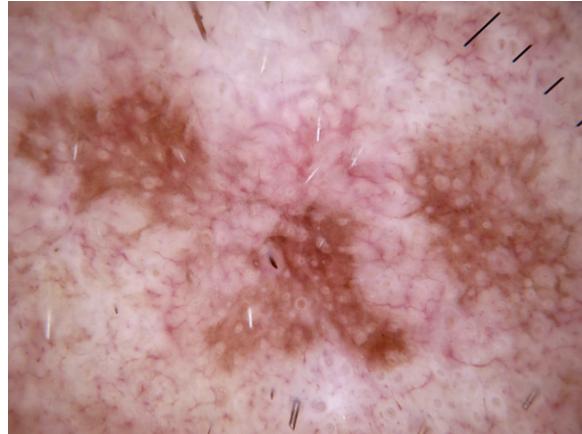


Figure 2.5: Actinic keratoses[2]

Symptoms

Actinic keratoses can have a variety of appearances. Among the signs and symptoms are:

- Scaly, dry, or rough patch of skin, usually less than an inch (2.5 cm) in diameter.[15]
- There is a flat to slightly elevated hump or patch on the outermost layer of skin.
- A rough, warty surface under some conditions.
- Color variations, such pink, brown, or red.
- Irritating, burning, seeping, or adhering.
- There are new lumps or patches on sun-exposed areas of the head, neck, hands, and forearms.

2.4.6 Intraepithelial carcinoma

One or more asymmetrical, scaly plaques with a diameter of several centimeters are the hallmark of intraepidermal SCC. Though they can sometimes be brown, they are typically orange-red in color.

While intraepidermal SCC can appear anywhere on the skin, it is typically identified on sun-exposed areas of the face, hands, ears, and lower limbs[16].

2.4.7 Basal cell carcinoma

One kind of skin cancer is called basal cell carcinoma. The skin's basal cells, which generate new skin cells as older ones age, are the source of basal cell carcinoma.

develop under nails, on the palms of hands, soles of feet, or inside the eye (ocular melanoma). These areas of the skin or body have never been exposed to the sun.

In 2023, there will likely be more than 18,200 new cases of melanoma diagnosed. Sixty-five is the average age at diagnosis[18] .

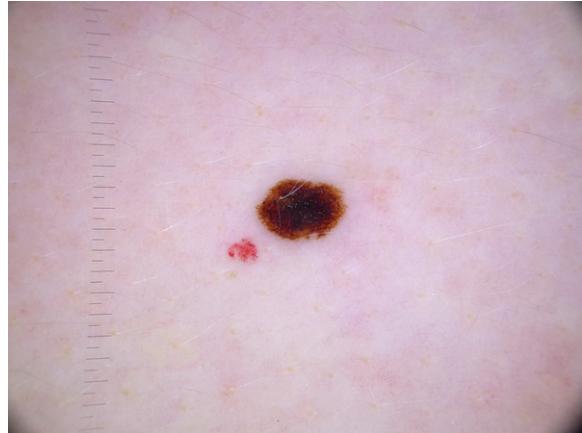


Figure 2.7: Melanoma[2]

Symptoms Although melanoma frequently exhibits no symptoms, the first indication is either a change in an already-existing mole or the emergence of a new lesion. These modifications may consist of:

- A mole may become blotchy, change color, or have multiple color tones.
- A mole may seem to enlarge.
- A mole may not be symmetrical, grow in height, or have an uneven shape.
- A raised spot on the mole may form.
- Bleeding or itching.

In the United States, there is an annual rise of 5.4 million instances of skin cancer. Melanomas are responsible for 75% of skin cancer-related deaths[19]. Since melanoma patients have a declining 5-year survival rate, early identification is essential. Using a novel disease taxonomy and algorithm, a computational method has been devised to proactively track skin lesions and spot cancer early[20].

2.5 Conclusion

It is essential for prompt diagnosis and efficient treatment to comprehend and identify a variety of skin conditions, including Actinic keratoses, Bacterial cell carcinoma, Dermatofibroma, Actinocytotic nevi, Intraepithelial carcinoma, Basal cell carcinoma, and Melanoma. The significance of routine skin exams and obtaining medical attention for any suspicious changes cannot be overstated, as certain disorders may be innocuous while others represent serious health hazards. Remaining crucial to reducing potential problems and enhancing patient outcomes are early identification, accurate diagnosis, and suitable therapy. The frequency and burden of these disorders can also be decreased by public awareness campaigns and education about skin health and disease prevention. People can enhance their self-defense and general well-being by adopting a proactive stance towards skin health.

Chapter 3

Background Study and Literature Review

3.1 Introduction

The classification of skin diseases is an area with a lot of ongoing research. A comprehensive review of previous research pertaining to the thesis study is provided in this chapter. This includes the essential techniques that are really useful for our work. Image processing methods including K-Nearest Neighbor, Naïve Bayes, and Support Vector Machine (SVM) classifiers have been employed by numerous researchers. In the research community, also deep learning methods including convolutional neural networks, Deep CNNs, Multi-Scale CNNs, Back Propagation Neural Networks, Mobilenets, InceptionV3, and Deep Residual Networks are gaining popularity and attention in the research community. This chapter concludes with a review of the literature that includes the results of current and well-known studies.

3.2 CNN Architecture

A convolution tool that, through a process known as "feature extraction," divides and recognizes the image's numerous features for analysis. The feature extraction network is made up of numerous pairs of pooling or convolutional layers. a fully connected layer that makes use of the convolution process' output and forecasts the image's class using the features that were previously extracted. The goal of this CNN feature extraction model is to minimize the amount of features in a dataset. It generates new features that are a summary of the preexisting features found in the initial feature collection. The CNN architecture diagram illustrates the numerous layers that make up CNN.[21]

3.2.1 Convolutional Layer

It applies a convolutional mathematical operation between the input image and a kernel, which is another name for a filter. The convolutional layer creates a feature map by swiping the filter across the input image and calculating the dot product between the filter and small parts of the image. The edges, corners, and textures of the input image are among the significant patterns and structures that this feature map captures. Convolutional layers, as opposed to completely linked layers, maintain the spatial relationships between pixels, which makes them ideal for computer vision and image identification applications. Subsequent layers in the network receive the output of the convolutional layer for additional feature learning and abstraction. CNNs are useful in a variety of image-related tasks because convolutional layers allow them to effectively extract hierarchical representations of visual information from input images.[21]

3.2.2 Pooling Layer

Convolutional neural networks (CNNs) use the pooling layer to shrink convolved feature maps, which lowers computational costs by reducing inter-layer connections. It works independently on every feature map, summarizing data using techniques such as Max Pooling, which chooses the largest element, Average Pooling, which finds the mean, and Sum Pooling, which adds up the components. It acts as a link between the fully connected and convolutional layers, allowing for independent feature recognition and feature generalization. In addition to improving network efficiency, this improvement lowers computational complexity, which improves the CNN's capacity to identify relevant characteristics and correctly classify input data.[21]

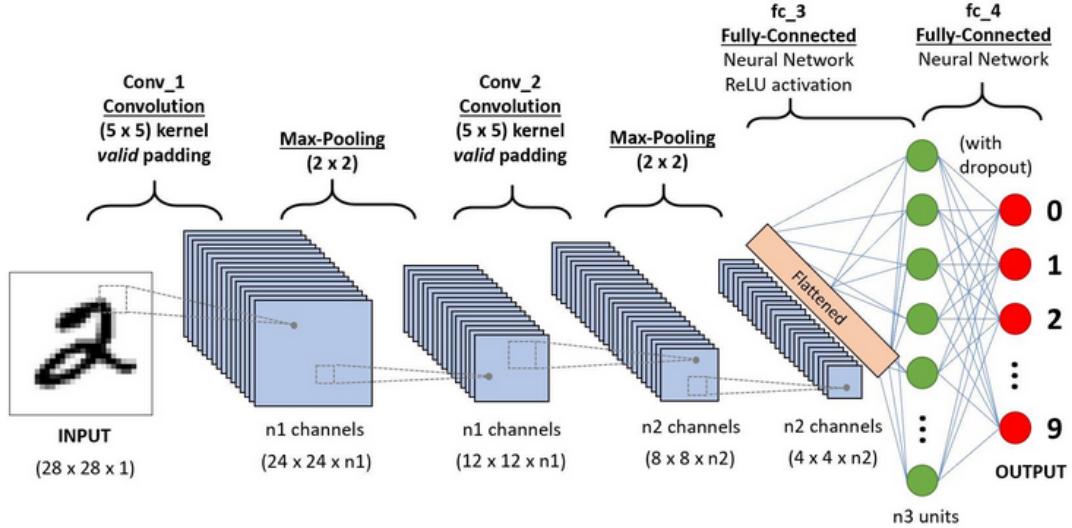


Figure 3.1: Convolutional neural networks[22]

3.2.3 Fully Connected Layer & Activation functions

In a CNN, neurons are connected across layers by the Fully Connected (FC) layer, which usually comes before the output layer. The classification process begins when input photos are flattened and sent via FC layers for mathematical computations. By improving model performance, several FC layers lessen the requirement for human supervision. Neuron activation and non-linearity are determined by activation functions, which are important. Popular functions such as tanH, ReLU, Softmax, and Sigmoid have specific uses; Softmax and Sigmoid are preferred for multi-class and binary classification, respectively. Activation functions give the CNN model more complexity and accuracy by determining neuron activation depending on the significance of the input. This allows for prediction through mathematical operations.[21]

3.3 Deep CNN Architecture

Deep Convolutional Neural Networks (CNNs) improve similarity detection by decomposing images into features, which are represented by MxM matrices. CNNs efficiently filter images and create feature maps by using convolution to search for feature matches across various places. In order to ensure mathematical stability, non-linear activation functions such as ReLU enable CNNs to learn intricate patterns by substituting zeros for negative values. The filtered images are subsequently downsampled by pooling layers, which lessens computational strain while

maintaining important information. The feature maps are ready for additional processing in completely connected layers after this series of steps.[23]

The flattened feature maps are supplied into fully connected layers after convolution and pooling, where they are converted into signals or votes that indicate the connection strengths for each category. As a result, CNN can now categorize photos using the features it has learnt. New images are inferred by passing through the lower layers until they reach the fully linked layer, when the category with the highest votes is elected. In conclusion, CNNs effectively extract features and categorize images by utilizing convolution, non-linearity, pooling, and fully linked layers. This allows for reliable and accurate image recognition and analysis.[23]

3.4 Overall deep CNN model architecture

A ReLU activation function, dropout, maximum pooling layers, and three fully connected layers with 1,024, 1,024, and 512 nodes, respectively, come after each convolutional layer[24]. The Softmax function is used by the final output layer to carry out three classifications. Convolutional neural network (CNN) and rectified linear unit (ReLU) are terms used in this context.

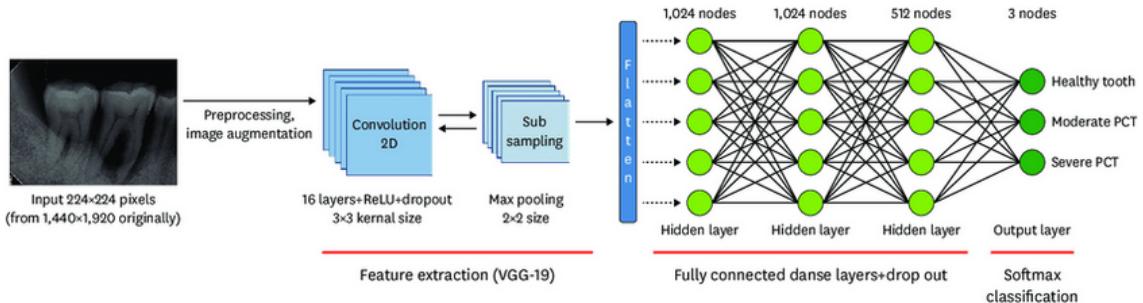


Figure 3.2: Overall deep CNN model architecture[25]

3.5 Deep Residual Network

A Convolutional Neural Network (CNN) architecture called Residual Network (ResNet) was developed to address the difficulty of efficiently training deep networks with hundreds or thousands of layers. The vanishing gradient problem, in which the gradient becomes smaller as it passes through more layers, was a challenge for conventional CNNs and prevented them from learning well. With the introduction of skip connections by ResNet, information could move

directly between layers, avoiding some of them. These skip connections—also referred to as identity mappings—assist in preserving the gradient signal, which makes deep network training more effective. ResNet accelerates training by compressing the network during the first training phase by omitting some layers. The entire network is then used for retraining, which enables the residual components to examine a larger feature space of the input image. ResNet topologies usually incorporate batch normalization and nonlinearity in between two or three layers that are skipped at a time. [26]

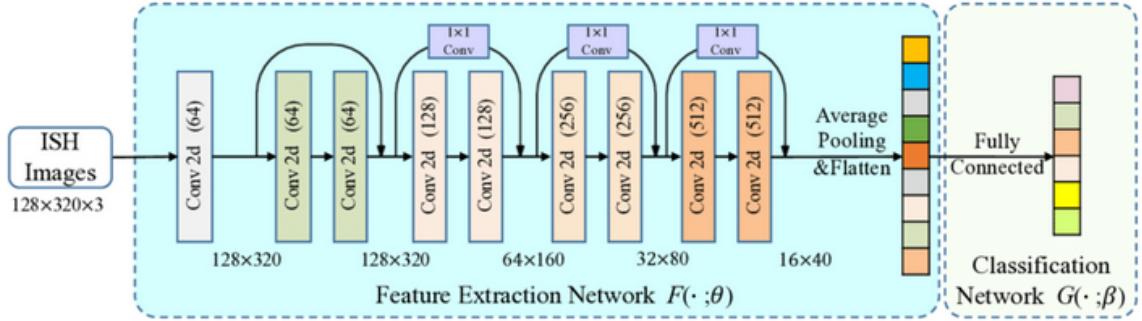


Figure 3.3: Deep Residual Network[27]

In addition, sophisticated ResNet variations like HighwayNets include the idea of skip weights, which dynamically decide how many layers to skip depending on the input data. With the help of this adaptive skipping mechanism, the ResNet architecture becomes even more versatile and adaptable, which boosts its performance in a variety of computer vision tasks. All things considered, ResNet's creative skip connections transformed deep learning by successfully resolving the difficulties involved in training deep neural networks, paving the way for the creation of extremely precise and effective models for computer vision applications.

3.6 Residual Blocks

An essential component of the ResNet architecture are residual blocks. Convolutional layers are stacked with batch normalization and nonlinear activation layers, like ReLu, in earlier systems like VGG16[28]. This technique only requires a modest number of convolutional layers; for VGG models, the maximum is about 19. Later studies, however, found that CNN performance could be greatly enhanced by adding more layers.

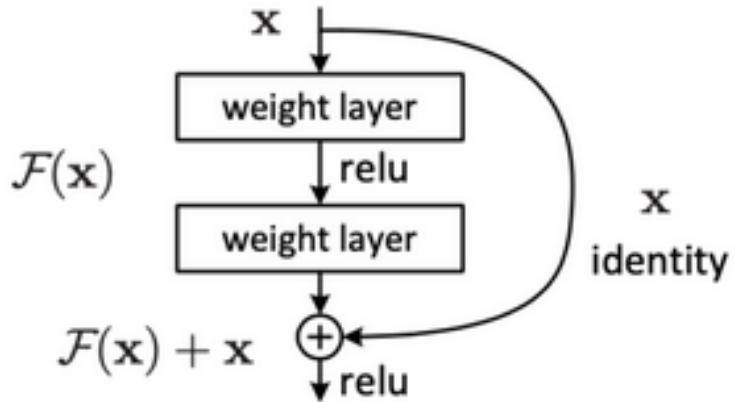


Figure 3.4: Residual Blocks

A fundamental part of the ResNet architecture, the residual block makes use of skip connections to enable smoother gradient flow during backpropagation. This method, which can be written in Python code as $\text{output} = F(x) + x$, enables deep networks to scale to hundreds of layers without adding more computing burden. Beyond ResNet, it has been widely incorporated into other neural network architectures, including UNet and Recurrent Neural Networks (RNN).[26]

3.7 Fully Convolutional Residual Network

The fully convolutional neural network architecture employed in this work is shown in the upper panel. The upper labels designate the many types of layers that each color represents. The bottom labels additionally provide information about the convolutional layers' kernel sizes. The input and output layers are represented by black layers. Lower panel: a residual block's inside structure.

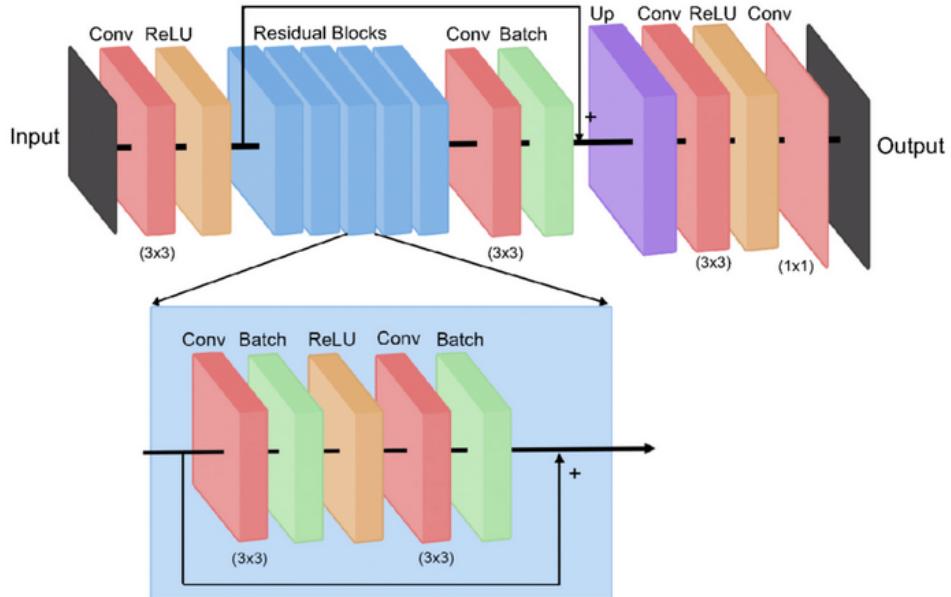


Figure 3.5: Fully Convolutional Residual Network[29]

3.8 Multi-Scale CNN

Convolutional Neural Networks (CNNs) work by breaking down input photos into a hierarchical pattern of visual elements, starting with pixels and working their way up to more complex features such as full faces and facial parts. Stage-by-stage alternating convolution and pooling procedures are used to accomplish this decomposition. To minimize dimensionality, the image is first convolutioned using learnt filters, and then it is pooled. This procedure is repeated in later stages with the addition of filters and pooling to cover greater image areas and process more intricate patterns.[30]

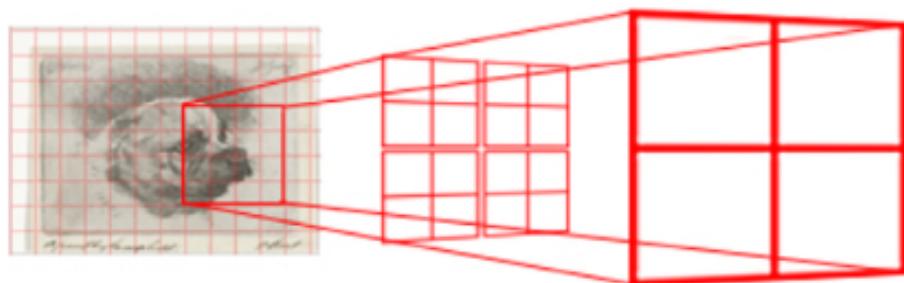


Figure 3.6: Output of 4 filters of Multi CNN Architecture

The size of the spatial output map is determined by the amount of subsampling in the net-

work architecture, which also affects the spatial extent of visual patterns analyzed by each stage. Larger input areas can be described by a deeper network that performs more subsampling operations. The relative amount of the input that each output describes diminishes as the size of the spatial output map grows with the size of the input. Smaller sections describe minute details, whereas larger portions capture broader elements. This has an impact on the characteristics of filters.

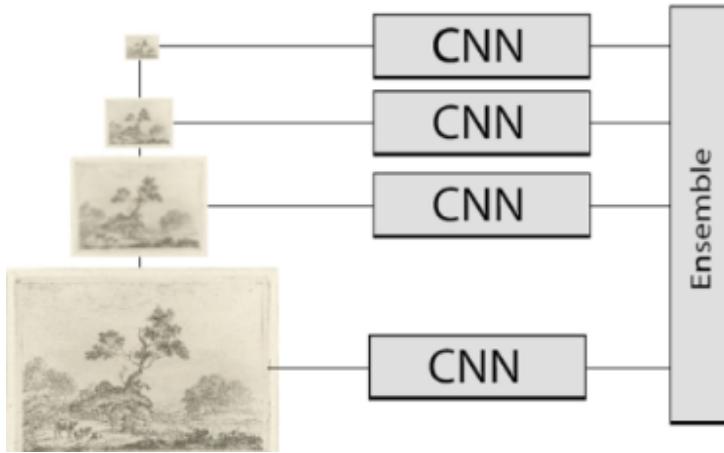


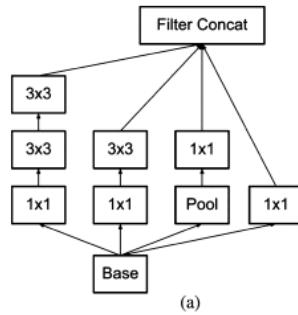
Figure 3.7: Visual representation of multi-scale cnn model

CNNs need multiple phases to acquire fine-to-coarse visual properties in order to examine images thoroughly. Information loss is avoided by using a deep network with multiple stages to accomplish gradual subsampling. Nevertheless, there are issues with training data availability and processing needs when networks get more sophisticated. Other strategies, such splitting up the work across specialized CNNs and merging predictions, provide workable answers to these problems.[30]

Using a multi-scale picture representation, the multi-scale CNN put forth here develops scale-variant and scale-invariant features by assigning distinct CNNs to each scale. A Gaussian pyramid is used to construct this representation, with each level after the first comprising smoothed and down-sampled representations of the earlier levels. While smoothing decreases duplication between neighboring pixels by conveying comparable information, downsampling is not required for the generation of pyramids.

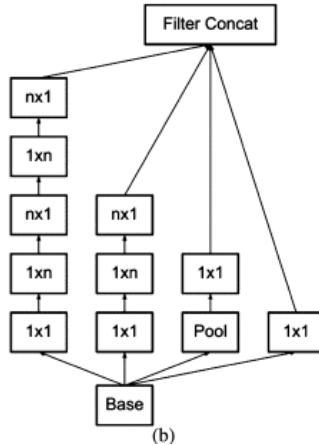
3.9 Inception v3

Inception-v1 splits a few 7×7 and 5×5 convolutional layers into multiple subsequent 3×3 convolutional layers[31]. By using a similar idea, Inception-v3 splits the symmetric $n \times n$ convolutional layers into two asymmetric layers, or $1 \times n$ and $n \times 1$ convolutional layers, and further reduces the convolution kernel size[32]. As an example, the 3×3 convolutional layer is split into 1×3 and 3×1 convolutional layers, which reduces the number of parameters from nine to six. Asymmetrical splitting like this can lead to more feature variation. This leads to the creation of three new Inception modules.[33]



(a)

Figure 3.8: (a)[34]



(b)

Figure 3.9: (b)[34]

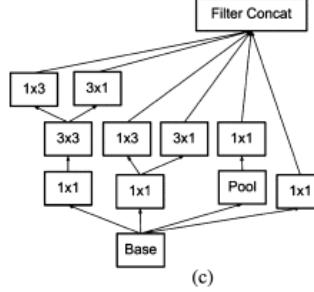


Figure 3.10: (c)[34]

Combining the three Inception modules from Fig. 3.18 yields Inception-v3. Then, as Table 3.4 illustrates, BN is applied to all convolutional layers and fully connected layers of the auxiliary classification network, factorizing the first 7×7 convolutional layer into three 3×3 convolutional layers. The network increases categorization accuracy even more.

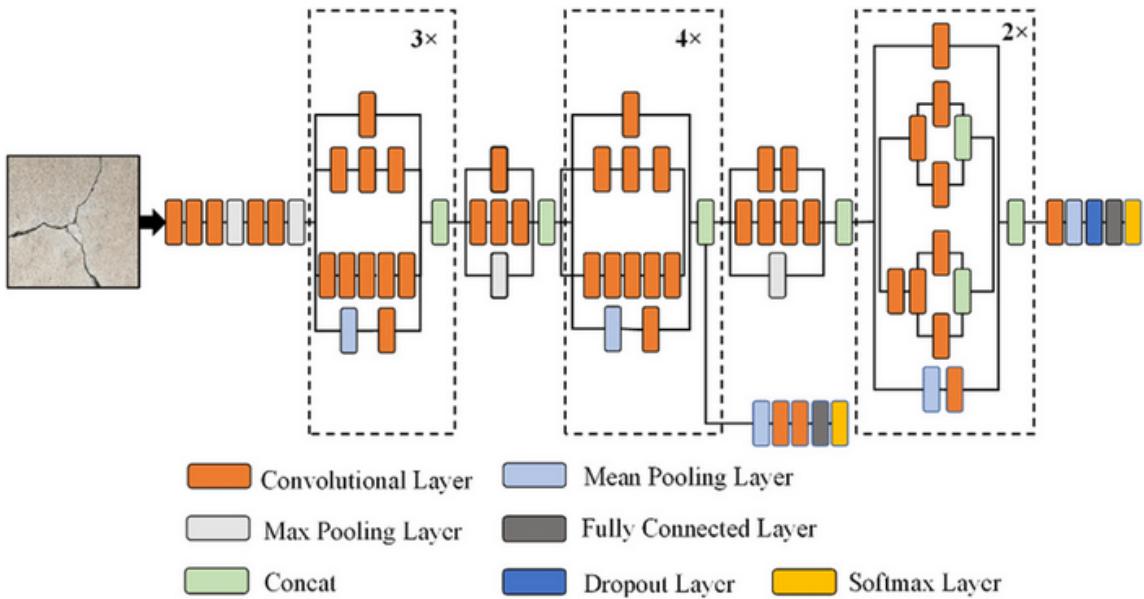


Figure 3.11: Inception v3[34]

3.10 Mobilenet

MobileNet is a popular deep convolutional neural network (CNN) used for computer vision applications such as picture classification and segmentation. In order to minimize latency and maximize compatibility for mobile deployment, depthwise separable filters play a vital role in its design. The two main components of the MobileNet design are pointwise and depthwise convolutions. The first step in feature extraction is depthwise convolutions, in which every input

channel is handled separately. These features are then combined using 1x1 pointwise convolutions. By separating the extraction and combining of features, calculation time, expense, and model size are reduced.[35]

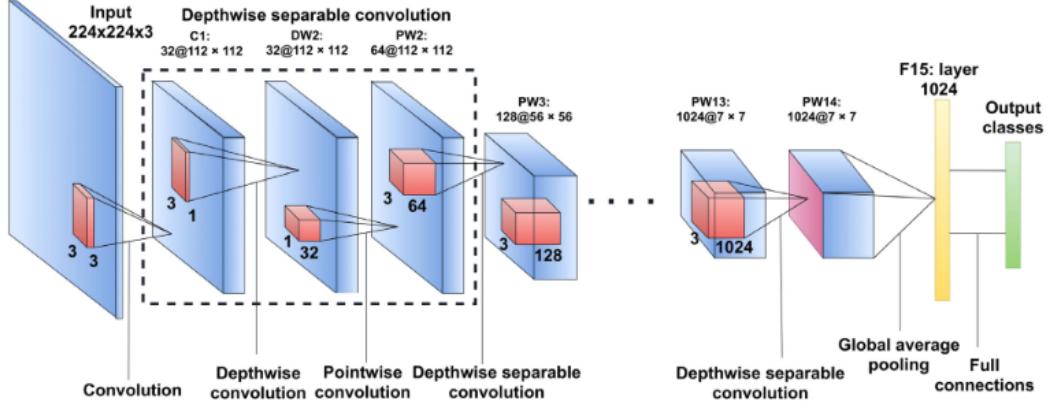


Figure 3.12: Mobile Net Architecture (Depthwise and Pointwise Convolution)[35]

Standard convolutional layers are not the same as depthwise convolutional layers architecturally. When an input feature map with dimensions of $DF \times DF \times M$ is fed into a normal convolutional layer, it outputs feature maps with dimensions of $DG \times DG \times N$, where DF and DG stand for input and output image dimensions, M for input depth, and N for output depth. The depthwise convolution kernel, abbreviated as \bar{K} , has dimensions $DK \times DK \times M$ for depthwise convolution layers. Each channel is computed independently by the depthwise convolution, yielding a computational cost represented by $DK \times DK \times M \times DF \times DF$. The corresponding channel in the input feature map is subjected to each filter in the depthwise convolution kernel, resulting in the corresponding channel in the output feature map.[35]

3.11 Support Vector Machine

One of the best supervised learning techniques that is readily available is support vector machines (SVMs). It is a kernel-based supervised learning method for binary classification tasks. It splits the two groups using a kernel function that comes from the training data set. The goal is to develop a classifier that works well on unseen samples and offers excellent generalization.

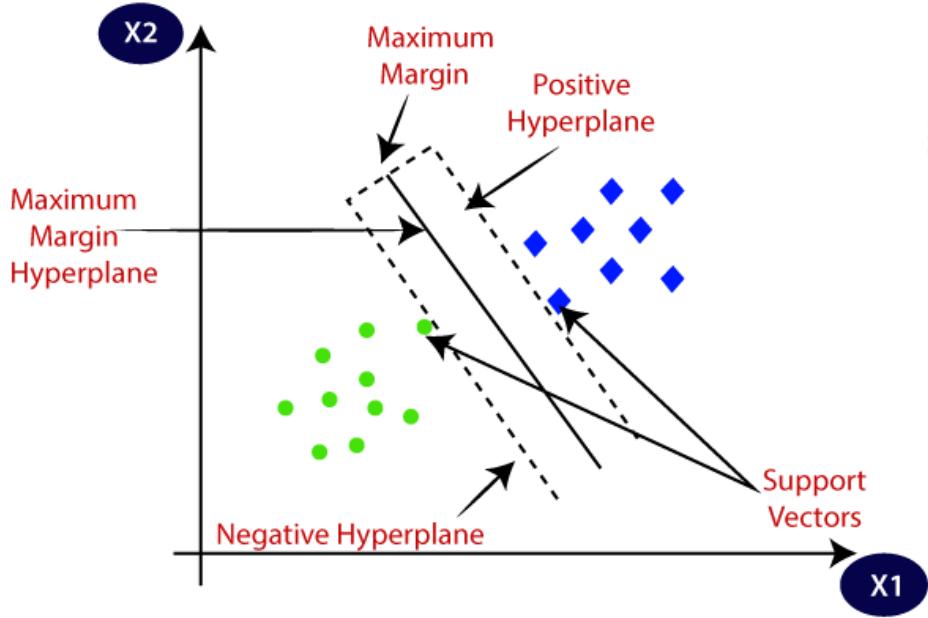


Figure 3.13: Support Vector Machine[36]

To help create the hyperplane, SVM chooses the extreme vectors and points. Since these extreme circumstances are referred to as support vectors, the method is known as a support vector machine. Looking at the following diagram, which uses a decision boundary or hyperplane to divide two categories into two groups.[37]

3.12 KNN

KNN is a popular non-parametric machine learning algorithm that uses supervision to learn from data. It is applied to classification as well as regression. The primary applications of the KNN algorithm are recognition, classification, and handwriting detection. The nearest neighbor point (k) in a KNN is the one that is closest to or accountable for the testing data class. A few formulas can be used to calculate the distance between the testing data and the nearest training class.[38] For example, the Manhattan distance , the Euclidean distance , the Mankowski distance , and the Hamming distance.

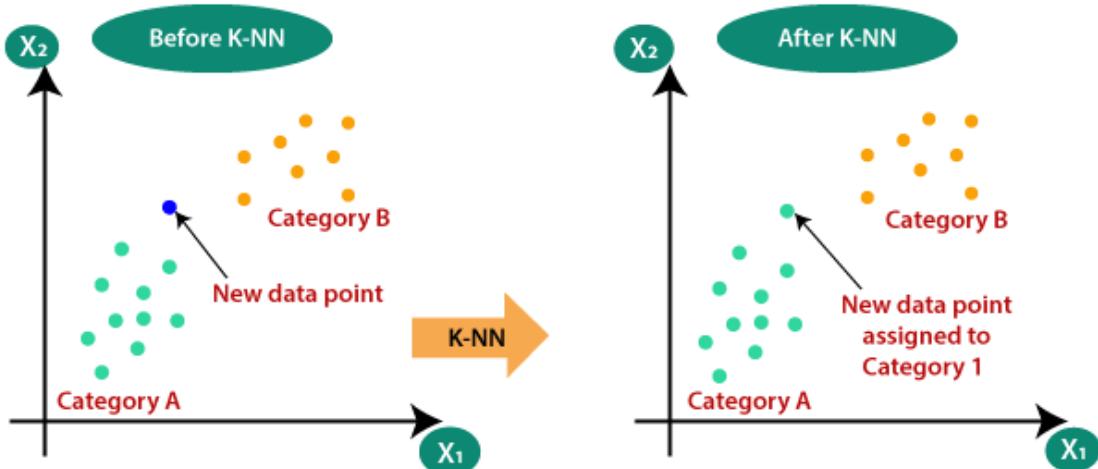


Figure 3.14: KNN[39]

3.13 Naïve Bayes Classifier

The Naive Bayes classifier is a probabilistic classifier that relies on the Bayes theorem and operates under the assumption of Naive (Strong) independence. Naive Bayes classifiers make the assumption that a variable's impact on a certain class is independent of other variables' values. We refer to this presumption as class conditional independence. Naive Bayes is frequently capable of carrying out more complex classification techniques. It works especially well when the inputs have a high dimensionality. We can utilize Naïve Bayes implementation to get output that is more competent than other algorithms. To build prediction models, one uses naïve Bayesian methods.[40] Bayes theorem: probability of B given A = probability of A and B / probability of A. The procedure counts the number of occurrences where A and B occur together in order to determine the likelihood of B given A.

3.14 Back Propagation Neural Network Algorithm

Three layers make up a feedforward neural network architecture: input, hidden layers, and output. The dimensions of the input data influence how many neurons are received by the input layer. The computational engine of the network is hidden layers, which are shielded from input and output. Each hidden layer's neurons apply an activation function, calculate the weighted total of the inputs from the layer before it, and then send the result to layers that come after.

Different numbers of hidden layers can be present in the network, enabling intricate feature extraction and representation.[41]

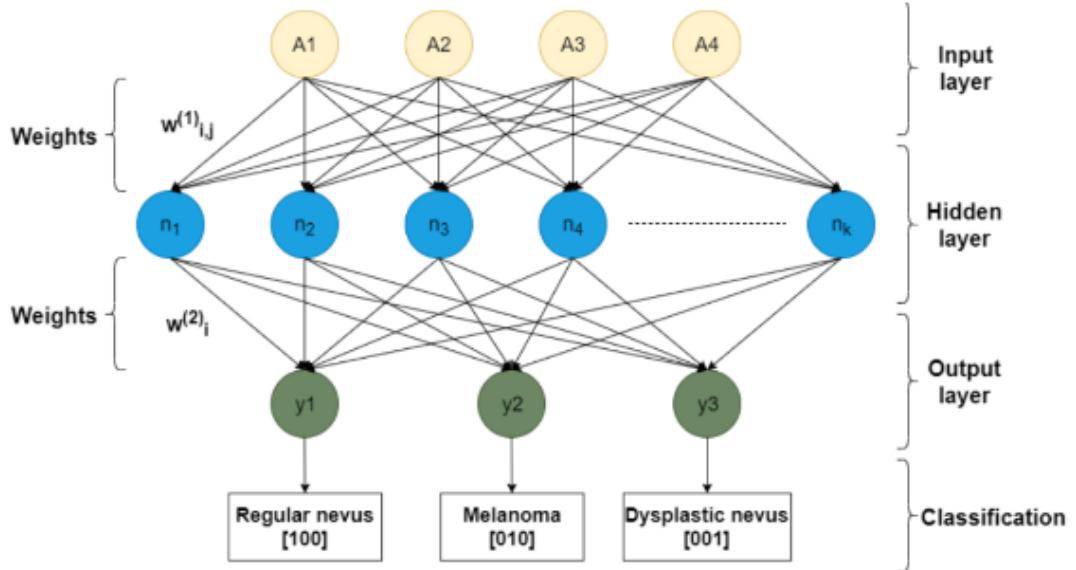


Figure 3.15: Back Propagation Neural Network[41]

Based on the inputs that have been processed, the output layer, which is the last part, produces the network's output. The number of potential output categories in this layer is correlated with the number of neurons. Adjacent layer neurons are fully coupled, and the weights between the neurons define the connections. Through iterative training procedures, the neural network can learn by modifying these weights in response to errors seen in the output. This allows the network to perform better overall.[41] The two main stages of the feedforward neural network's operation are feedforward and backpropagation. In the feedforward phase, when input data flows through the network, each hidden layer adds non-linearity by calculating the weighted sum of its inputs and then sending the result via an activation function. Until an output prediction is produced at the output layer, this process is repeated. The error of the forecast is then calculated and spread backward through the network during the backpropagation phase. The network continually modifies its weights using a gradient descent optimization approach to reduce this error, allowing for ongoing improvement in prediction accuracy during training iterations.

3.15 Literature Review

Skin conditions caused by air pollution, such as eczema, melanoma, psoriasis, and actinic keratosis, are common throughout the world and affect 7.9–60% of people[42]. Medical technology has advanced, yet diagnosis is still expensive and scarce. This paper suggests a two-stage method for early and affordable skin disease identification and classification that makes use of image processing techniques and a Naive-Bayes classifier.

In order to overcome contrast and variation issues, this paper presents an automated melanoma recognition approach using dermoscopic images using deep convolutional neural networks (CNNs). In order to achieve accurate classification with little training data, it integrates CNN and FCRN.[43] Specificity is improved by segmenting using residual networks and then classifying neural networks; this is assessed using performance metrics and dataset analysis.

This study uses Convolutional Neural Network (CNN) architectures, such as Xception and MobileNet, to propose a robust method for diagnosing skin illnesses. The proposed method achieves higher classification accuracy rates of 97.00% and 96.00%, respectively, by applying augmentation and transfer learning strategies.[35] In addition, a web-based architecture is implemented to provide real-time disease identification.

Scale-variant representations are developed using a standard CNN that has not been trained with any additional data[30]. As a result, a CNN cannot handle scale-variant features at different scales; it can only recognize the features it was trained on, at the scale it was trained on.[30] Introducing different scales to the CNN during training is a simple way to get around this restriction; this technique is known as "scale jittering".

With dermoscopic images, this study uses deep learning algorithms to detect four prevalent skin illnesses with an accuracy of $87.25 \pm 2.24\%$.[44] Semantically describing diagnosis scenarios improves algorithm accuracy and decision-making. The study emphasizes how human judgment and computer algorithms work together to diagnose dermatological conditions.

Over 90% of people worldwide have skin illnesses at some point in their lives[45]. The frequency of skin diseases is higher in men and rises with age. This work uses a feed-forward arti-

ficial neural network (ANN) system for disease identification and image processing for feature extraction with a focus on dermatological healthcare[46]. The framework detects six different forms of skin disorders with 95% accuracy[46].

Chapter 4

Methodology

4.1 Introduction

In this chapter, we'll extravagantly examine our work and our proposed strategy. In this chapter, the coordination the LSTM with the MobileNet V2 is clarified with an architecture. MobileNet V2 is utilized in classifying the sort of skin malady, and LSTM is utilized to improve the execution of the demonstration by keeping up the state data of the features that it comes over within the previous era of the image classification.

MobileNet V2 could be a CNN-based demonstration that's broadly utilized to classify pictures. The biggest advantage of utilizing the MobileNet V2 design is that the show needs comparatively less computational exertion than the ordinary CNN demonstration, which makes it appropriate for working over versatile gadgets and computers with lower computational capabilities.

4.2 Methodology

The proposed system for the viable suggestion includes different stages within the handle of classifying the sort of the skin disease is displayed. Within the introductory stage, the information are obtained and evaluated by the experts and specialists for the sort of illness for precise preparing of the demonstrate.

The moment stage of the system concerns the integration of the proposed MobileNet V2 with LSTM demonstrate. In this stage, the picture of the influenced locale is captured and bolstered as the input for the demonstrate, the highlights of the input picture are recognized for connecting the highlights with the prepared information for forecasts. The probabilities of the specific sort

of infections are approximated in this phase to decide the lesson of the infection.

Within the third stage, the classification result and the assessment of the demonstrate are performed. The malady classification likelihood decides the lesson of the illness, and the result of the forecasts are assessed against the different assessment measurements.

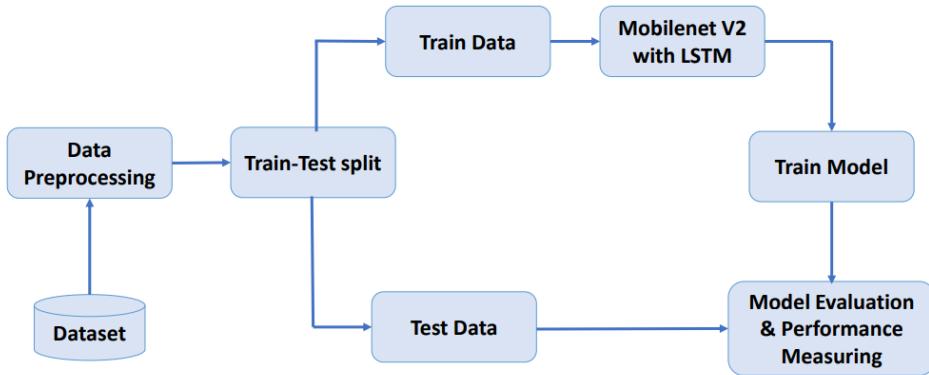


Figure 4.1: Proposed Model

4.3 Overview of Proposed Methodology

The method of categorizing the type of skin illness is divided into several stages in the suggested work. Our suggested method evaluates the type of disease using many photos and the suggested MobileNet V2 using the LSTM technique.

We began working with the dataset in the first step. The HAM10000 dataset is accessible for skin diseases. Preprocessing of the datasets was necessary. We have normalized the photos to improve their performance because the datasets are quite variable. After that, we used several data augmentation techniques to add diversity to the datasets because some skin illnesses only have a small number of photos.

We are focused on integrating the suggested MobileNet V2 with the LSTM model in the second stage of our study. During this stage, the impacted area's image is taken and supplied to the model as input.

In order to correlate the features with the trained data and make predictions, the features of the input image are detected. In this phase, the illness class is determined by approximating the

probabilities of the specific type of sickness.

The model's evaluation and the classification result are carried out in the third phase. The class of the disease is determined by the disease classification probability, and the database is updated with the results of the predictions after they have been assessed using a variety of assessment criteria.

4.4 Dataset Descriptions

The skin disease dataset, known as HAM10000, was taken from Kaggle and used as a benchmark database that was retrieved straight from the source[47]. The age, gender, and cell type metadata for the dataset are provided in a comma-separated values file (.CSV) format.

Over 10,000 dermatoscopic photographs from various individuals worldwide are included in this dataset.

In order to improve the model's accuracy and performance, the dataset also offers extra advice on how to deal with issues like overfitting and a lack of data. There are seven main kinds of skin issues in this dataset.

To improve the generalization of our model, the dataset is split into three sections: training data, validation data, and testing data, which comprise 85%, 5%, and 10% of the total. The ground truths connected to the training dataset are used to assess the model.

Our suggested model requires photos to have a goal size of 224×224 . The purpose of this study is to evaluate how well our suggested method detects skin cancer on dermatoscopic images.

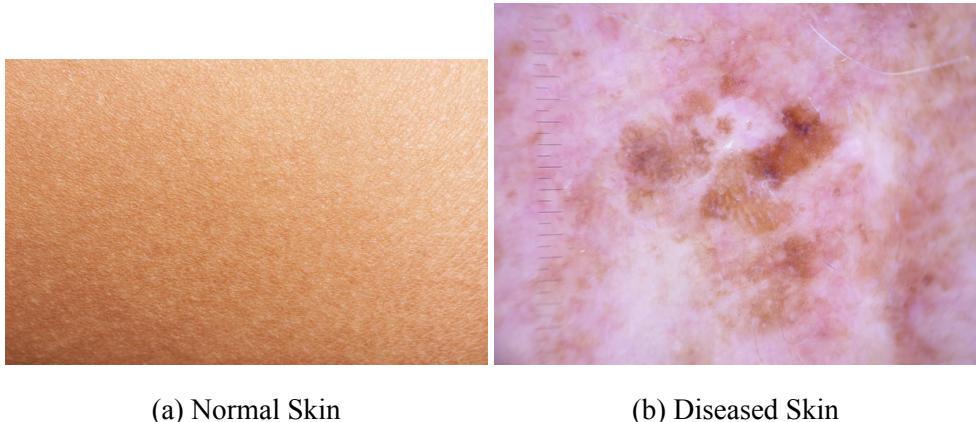


Figure 4.2: Picture of normal and diseased skin

Table 4.1: HAM10000 Dataset

Class	Number of Samples
Melanocytic nevi	6705
Melanoma	1113
Benign keratosis-like lesions	1099
Basal cell carcinoma	514
Actinic keratoses	327
Vascular lesions	142
Dermatofibroma	115

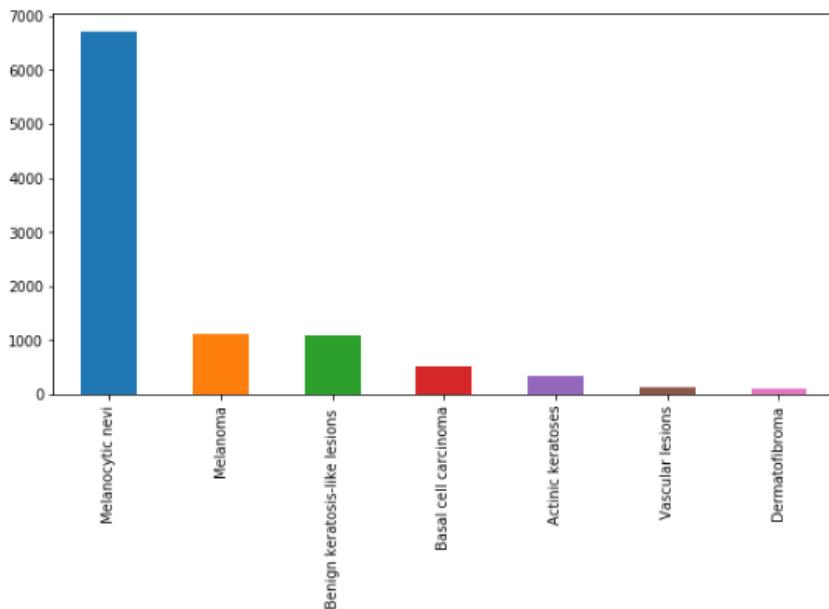


Figure 4.3: HAM10000

4.5 Dataset Preprocessing

Preprocessing data prepares it for additional analysis or modeling and is the first stage in the data analysis pipeline. Usually, it is completed prior to using any machine learning or statistical methods on the data. Because it guarantees data quality, boosts model performance, increases algorithm compatibility, lessens overfitting, increases interpretability, and conserves time and resources, preprocessing is crucial.

4.5.1 Normalization

Normalization is a common preprocessing step applied to data before it is fed into machine learning models. Model performance may improve and training time may be accelerated as a result. Scaling the input attributes to a comparable range is its aim.

Normalization is sometimes achieved by scaling the attributes to a range of 0 to 1, or by having a mean of 0 and a standard deviation of 1. It anticipates that the images will be in BGR format per OpenCV norm.

Prior to calculating the mean and standard deviation, each pixel value is normalized to the range [0, 1] by dividing it by 255.0.

4.5.2 Image resizing

Images in our dataset are not uniform. When working with datasets that contain photos of different sizes, scaling photos is a typical preprocessing step. Resizing is the process of changing an image's dimensions to a target size while maintaining its aspect ratio.

Using OpenCV, each image is loaded and scaled to a constant 224x224 pixel size.

4.5.3 Data Augmentation

To enhance the data for classes with fewer examples in order to balance the number of samples across the seven classes (cell types) in the training dataset.

By running this code, it augments the data for classes with fewer examples, balancing the number of samples among the various classes in the training dataset.

This helps avoid problems with class imbalance, which can have a detrimental effect on how well machine learning models perform, particularly when it comes to classification tasks. This method makes use of conventional data augmentation techniques like flipping, scaling, and rotation.

Table 4.2: Data Augmentation Techniques

Augmentation Type	Settings
Rotation range	20
Shift range along width	0.2
Shift range along height	0.2
Shear intensity	0.2
Zoom range	0.2
Horizontal Flip	True
Vertical Flip	True

4.6 Training Set & Validation Set

A training subset and a validation subset were created from the dataset. 89% of the total dataset is made up of the training subset, while the remaining 11% is validated. The total number of images present in the dataset is 10015.

Among the 10015 images, 8912 are used for training, and 1103 are used for validation. With the help of this split, the model is guaranteed to be trained on most of the dataset, which enables it to identify patterns and associations. The model modifies its parameters while it is being trained using the training data.

Enhancing the model's performance metric, which depending on the particular situation, may include accuracy, precision, recall, F1-score, or any other pertinent metric is the goal. The goal of the model is to determine the optimal configuration that optimizes the selected performance metric by fitting the training data into the model and modifying its parameters.

To get the best performance on the training set of data, the model's parameters are iteratively adjusted during the optimization phase. The model can now be evaluated with validation data following the training phase. The validation data includes information that the model did not observe during training, and its objective is to evaluate the model's capacity for generalization.

Table 4.3: Training Subset

Class	Number of Samples
Melanocytic nevi	5822
Melanoma	1067
Benign keratosis-like lesions	1011
Basal cell carcinoma	479
Actinic keratoses	297
Vascular lesions	129
Dermatofibroma	107

Table 4.4: Validation Subset

Class	Number of Samples
Melanocytic nevi	883
Melanoma	46
Benign keratosis-like lesions	88
Basal cell carcinoma	35
Actinic keratoses	30
Vascular lesions	13
Dermatofibroma	8

4.7 Experimental Setup

This experiment was performed on an online compiler named Kaggle. Kaggle provides a comprehensive platform that integrates data sets, coding environments, collaboration tools, and competitions all under one roof. The following environment served as the setting for the entire implementation:

System Configuration:

CPU: Intel(R) Core(TM) i3-8130U CPU @ 2.20GHz, 2208 Mhz, 2 Core(s), 4 Logical Pro-

cessor(s)

RAM: 4GB

System Type: 64 bit operating system

4.8 Libraries

NumPy, pandas, OS, matplotlib, pyplot, seaborn, cv2, tensorflow, sklearn, and torchvision are the libraries that we used in our model. For image processing and plotting, including creating graphs, charts, and tables, the Matplotlib, pyplot, and Seaborn libraries are utilized.

Path and directory operations on files, as well as file collection, are provided by the Shutil and os libraries. We import the torchvision and seaborn libraries to generate models such as the confusion matrix, ROC curve, and classification report. The most widely used libraries for array processing and data analysis are pandas and numpy.

4.9 MobileNetV2

The MobileNetV2 architecture consists of an initial fully convolution layer with 32 filters, which also includes 19 residual bottleneck layers. Using a 3×3 kernel, ReLU6 is utilized for non-linearity.

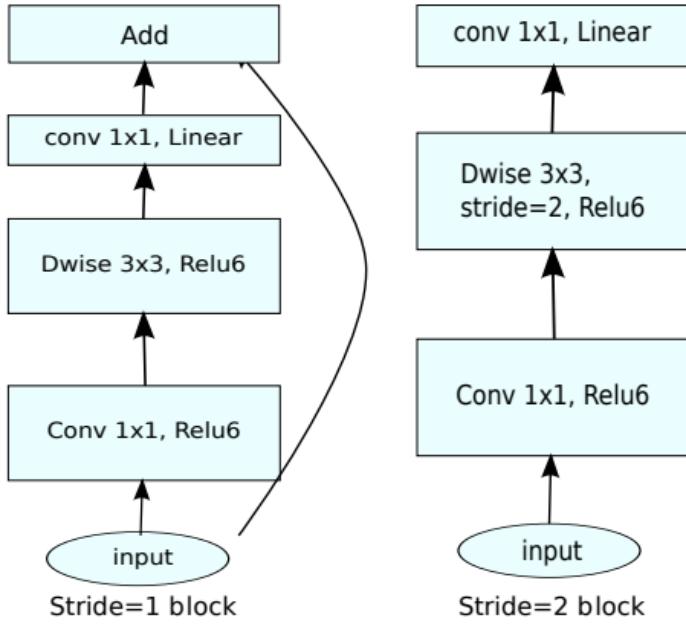


Figure 4.4: Mobilenet V2 Architecture[48]

During training, dropout and batch normalization are applied. Smaller networks perform better at expansion rates between 5 and 10, which provide almost comparable performance curves. To the supplied tensor size, an expansion factor of 6 is applied.

By employing the input picture resolution and width multiplier as tunable hyperparameters that can be changed based on desired accuracy/performance trade-offs, we can adapt our architecture to different performance points. Our main network (width multiplier 1, 224×224) requires 3.4 million parameters and 300 million multiply-adds to compute. We investigate the trade-offs between performance for input resolutions ranging from 96 to 224 and width multipliers from 0.35 to 1.4. The model size varies from 1.7M to 6.9M parameters, while the network computational cost spans from 7 multiplies adds to 585M adds.

There is a small variation in the implementation: we apply the width multiplier to all layers except the final convolutional layer when the multiplier is less than one. Smaller models perform better as a result.[48]

Input	Operator	<i>t</i>	<i>c</i>	<i>n</i>	<i>s</i>
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

Figure 4.5: Mobilenet V2

Every line denotes a series of *n* times repeated identical (modulo stride) layers. The number *c* of output channels is the same for every layer in the same sequence. Every sequence has a stride *s* for the first layer and a stride 1 for every subsequent layer. 3 x 3 kernels are used in all spatial convolutions[49].

4.10 LSTM

Long short-term memory (LSTM) is a gradient-based technique that tackles the vanishing error problem. More than 1,000 discrete time steps can have minimum time delays that an LSTM can learn to bridge. Constant error carousels (CECs), which maintain a continuous error flow inside particular cells, are used in the solution. Multiplicative gate units manage cell access; they determine when to allow access[49].

The Logic Behind LSTM

The LSTM cycle involves a single-time step where the cell decides whether to remember previous timestamp information, learns new information, and then passes updated information to the next timestamp.

An LSTM unit consists of three gates: Forget gate, Input gate, and Output gate. These gates control information flow in and out of the memory cell, forming a layer of neurons in a traditional feedforward neural network.

An LSTM has a hidden state, just like a basic RNN, where $H(t-1)$ is the hidden state of the times-

tamp that was previously recorded, and H_t is the hidden state of the timestamp that is currently recorded. Furthermore, the cell state of an LSTM is denoted by $C(t-1)$ for the past timestamp and $C(t)$ for the present timestamp, respectively. In this case, long-term memory refers to the cell state and short-term memory to the hidden state[50]. Take a look at this picture.

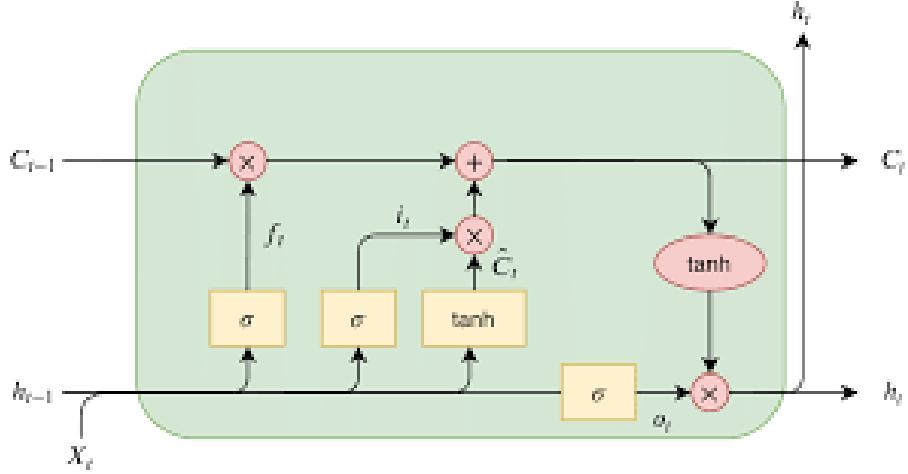


Figure 4.6: LSTM Architecture[51]

Input (c_{t-1}, h_{t-1}, x_t): x_t is the input of the t th state of the model. c_{t-1} and h_{t-1} are the outputs of the previous layer. h is the result of the tanh function[52].

Output (c_t, h_t): c cell state and h are hidden state. f_t, i_t and o_t are the forget, input, and output gates, respectively[52].

The modern highlight of LSTM compared to RNN is the transformation line from c_{t-1} to c_t . This makes a difference the vital data to be sent to begin with and utilized. In this manner, LSTM can carry data remotely (long-term memory) and combine it with the specials acquired from RNN (brief memory). The LSTM include was reasonable for our input information[52].

Memory Blocks

The CEC's backflow remains constant in a neural network, but it is connected to other units, causing conflicting weight updates. LSTM extends the CEC with input and output gates connected to the network input layer and other memory cells. This results in a memory block, with input gates controlling signals from the network to the memory cell and output gates controlling access to the memory cell. The basic function of multiplicative gate units is to allow or deny access to constant error flow through the CEC

4.11 Mobilenetv2 with LSTM

The element that is most frequently utilized with recurrent neural network topologies is LSTM. On pattern estimation challenges, it can rely on its learning sequence. Memory cells, which are part of the abstract LSTM layer module, control memory blocks. They consist of an input and output gate, a forgotten gate, and a window connection[2]. The persistent abstract LSTM memory module's activation function is described by the computations. The memory is included in the LSTM module.

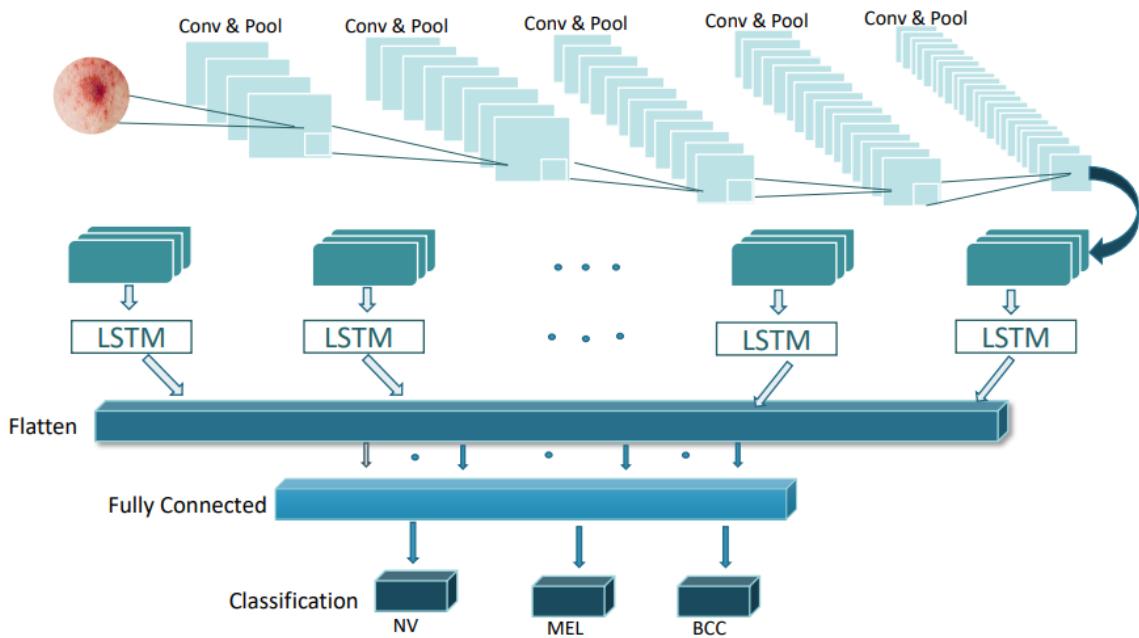


Figure 4.7: Mobilenet V2 with LSTM[2]

In Figure, the integration model is displayed. The entire architecture of the MobileNet V2 with the LSTM model, which consists of an LSTM component coupled to the model's flattening layer along with a combination of set of convolutions and max pooling layers, is shown in Figure.

the fully connected layer, which uses training to correlate the detected features with the pre-existing data. Lastly, the softmax layer, which calculates the likelihood of different disease groups[2].

4.12 Pre-Declaration

4.12.1 Dataset Division

We have separated our dataset into training and validation categories in order to assess our model. The validation dataset was used to improve the model's parameters and to provide predictions, while the training dataset was utilized to fit the model. The total dataset was split 89:11 between the training and validation halves.

4.12.2 Batch Size

The total number of samples in a particular batch is its batch size. The complete dataset has been divided into many batches for better and more efficient training. 32 is selected as the batch size for our classification jobs.

4.12.3 Epoch Size

Iterations are generally referred to as epochs. In a single epoch, the model runs the entire dataset both forward and backward. In a single period, one or more batches may be included. We have selected an epoch size of 30 for our study. Ten epochs are used to fine-tune the model. Total epoch count, encompassing both the initial training and fine-tuning phases. Because transfer learning requires fewer iterations to converge, the epoch size is not particularly huge.

4.13 Compile the Model

To put together the model, three factors are required: an optimizer, a loss, and metrics. In this instance, our recommended model was trained using the RMSprop optimizer, which has a learning rate of 0.00001.

Similar to stochastic gradient descent (SGD) with momentum, RMSprop is an optimization approach that normalizes gradients based on their magnitude in recent runs.

We used binary cross-entropy as our loss function. Binary cross-entropy is a loss function used in binary classification problems. It is often referred to as log loss or logistic loss. It calculates the difference between the actual binary labels and the projected probabilities in two probability distributions. When the anticipated likelihood is low and the actual label is 1, the binary cross-entropy loss penalizes inaccurate predictions more severely, and vice versa. Because of this, it is appropriate for binary classification tasks

Table 4.5: Validation Subset

Parameters	Values
Optimizer	RMSprop
Learning Rate	0.00001
Loss Function	$\text{binary}_{crossentropy}$

4.14 Conclusion

This chapter has addressed the outline of our suggested methodology. Included is a synopsis of the datasets that we used in our study. Our research methodology's specifics have been disclosed in great depth. We tested our proposed model architecture for Mobilenet V2 with the LSTM model.

Chapter 5

Result Analysis

5.1 Introduction

This chapter presents the experimental data, a performance analysis of our suggested CNN and RNN model combination, and a comparison with two other models. We evaluate our suggested models using several measures, such as accuracy, precision, recall, F1-score, and confusion matrix. Additionally discussed have been the accuracy and loss curves during the training and validation procedures.

5.2 Evaluation Metrics

Evaluation metrics are quantitative measurements used to evaluate a statistical or machine learning model's performance. These metrics assist compare various models or algorithms and give information about how well the model is doing. It is necessary to evaluate the generalizability, prediction accuracy, and universal quality of a machine learning model.

Various standards of evaluation are provided based on the particular machine learning task. Several often used assessment matrices are accuracy, precision, recall, R-squared, adjusted R-squared, F1-score, mean absolute error, and mean squared error.

Reason for evaluating metrics:

Metrics for evaluation are important for the following reasons:

- **Evaluating Model Performance:** They offer numerical values that indicate how well a model performs on a certain activity, which helps to identify its advantages and disadvantages.

vantages before it is perhaps put into use in a production setting.

- **Comparing Models:** Metrics make it possible to compare many models that have been trained on the same dataset, which helps determine which model performs the best when it comes to particular parameters like accuracy, precision, or recall.
- **Adjusting Hyperparameters:** Metrics are used to adjust hyperparameters in order to maximize the model performance. Scientists in Data Field can improve model efficacy by modifying hyperparameters like learning rate and epochs based on metric outcomes.
- **Monitor the performance of a model:** Metrics help track a model's performance over time in order to identify any declines brought on by idea drift or variations in the distribution of data. This enables the early detection of issues and the timely implementation of corrective measures.
- **Finding Overfitting:** Overfitting occurs when a model performs well on training data but poorly on unknown data. Evaluation metrics are useful in identifying overfitting. Overfitting can be found and fixed by comparing performance on training and test sets.

5.3 Classification Report

Each classification model evaluates the performance of a specific test set of data using a confusion matrix.

	Actual 0	Actual 1
Predicted 0	True Negative (TN)	False Negative (FN)
Predicted 1	False Positive (FP)	True Positive (TP)

- **True Positive (TP):** A positive prediction that comes true.
- **True Negative (TN):** A negative prediction that comes true.
- **False Positive (FP):** A positive prediction that isn't true.
- **False Negative (FN):** A negative prediction that isn't true.

5.4 Types of Evaluation Metrics:

- **Accuracy:** One of the most often applied assessment measures in classification issues is accuracy. It calculates the percentage of accurate forecasts among all the forecasts made.

It has the following definition:

The ratio of accurate predictions to total predictions determines accuracy.

It can be expressed mathematically as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

- **Precision:** The accuracy with which the model forecasts the desirable results is measured by its precision. Because a high precision suggests that most expected positives are real positives, it reduces the likelihood of false positives.

Precision can be expressed as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- **Recall:** Recall quantifies the model's capacity to recognize all pertinent positive examples. Recall is a measure of how few true positives the model misses.

The definition of recall in mathematics is:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- **F1-Score:** A popular metric for classification problems that aggregates recall and precision into a single number is the F1 score. When there is an imbalance in the classes, it is especially helpful.

The F1 score is computed using the following formula:

$$\text{F1 Score} = 2 * \left[\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \right]$$

At 1, F1 score is at its highest, the F1 score is at its highest, and at 0, it is at its lowest. It is a valuable indicator for assessing classification models because it strikes a balance between recall and precision, particularly in cases where the classes are unbalanced.

5.5 Result Analysis

CNN and ANN combined based networks are utilized in this study to detect skin diseases. The implemented code was run for 25 epochs and also for 10 epochs after fine tuning, during which time its accuracy and losses were tracked. Batch size was taken 32 and learning rate was 0.0001.

Table 5.1: Below some epoch and their values

No of epochs	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	0.6456	0.9635	0.6571	0.8753
5	0.6454	0.9639	0.6600	0.8616
10	0.6448	0.9670	0.6574	0.8728
15	0.6443	0.9700	0.6568	0.8791
20	0.6443	0.9692	0.6563	0.8778
25	0.6440	0.9703	0.6560	0.8791

After evaluating the models:

Evaluation Loss: 0.6585 Evaluation Accuracy: 0.8578

A graph showing the accuracy and loss during training and validation looks like this:

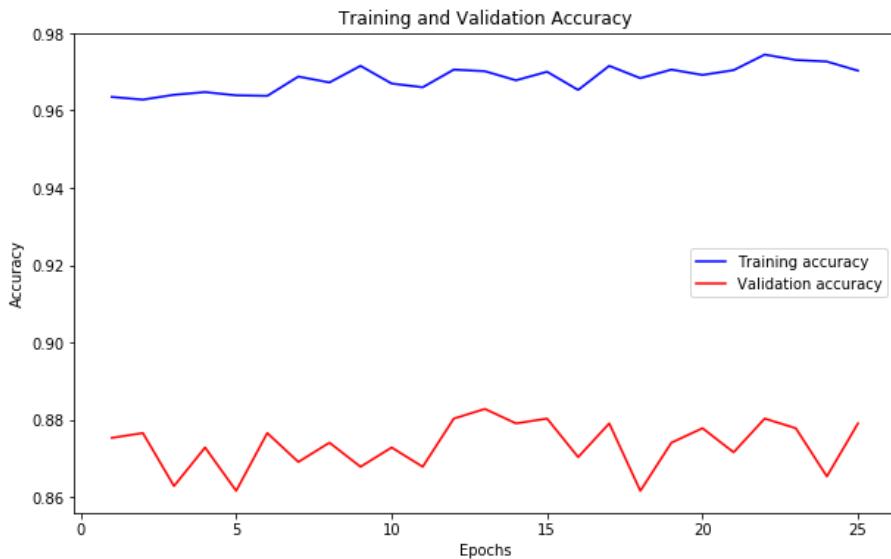


Figure 5.1: Training and validation Accuracy

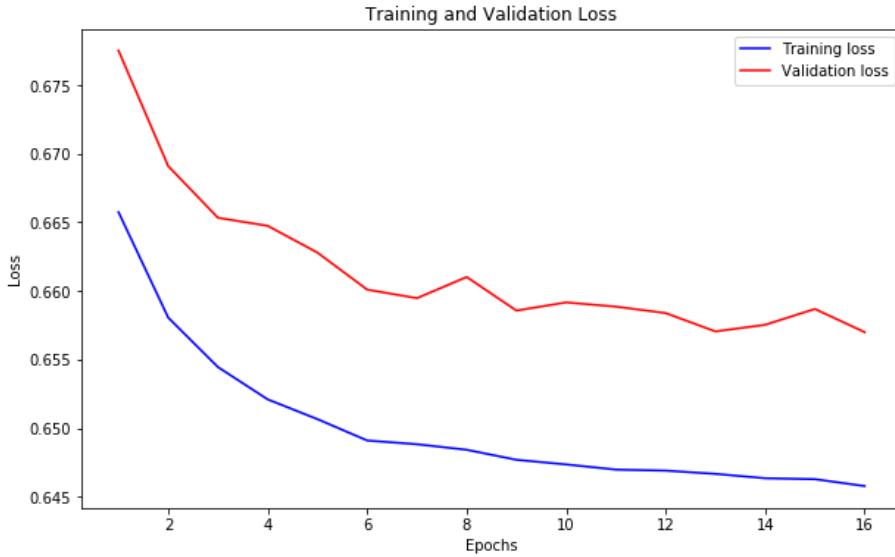


Figure 5.2: Training and validation loss

The training and validation loss, accuracy graphs show clear signs of overfitting. When a model is overfitted, it means that it has absorbed too much of the noise and fluctuations present in the training set. This happens when a model performs better than what is needed to solve a specific issue. Furthermore, this could be brought on by the model having too much training time.

Table 5.2: After Fine Tuning some epoch and their values

No of epochs	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
25	0.6436	0.97739	0.6578	0.8666
30	0.6436	0.9738	0.6566	0.8778
35	0.6434	0.9743	0.6561	0.8766
40	0.6428	0.9778	0.6559	0.8815

After evaluating the models:

Evaluation Loss: 0.6554 Evaluation Accuracy: 0.8802

A graph showing the accuracy and loss during training and validation looks like this:

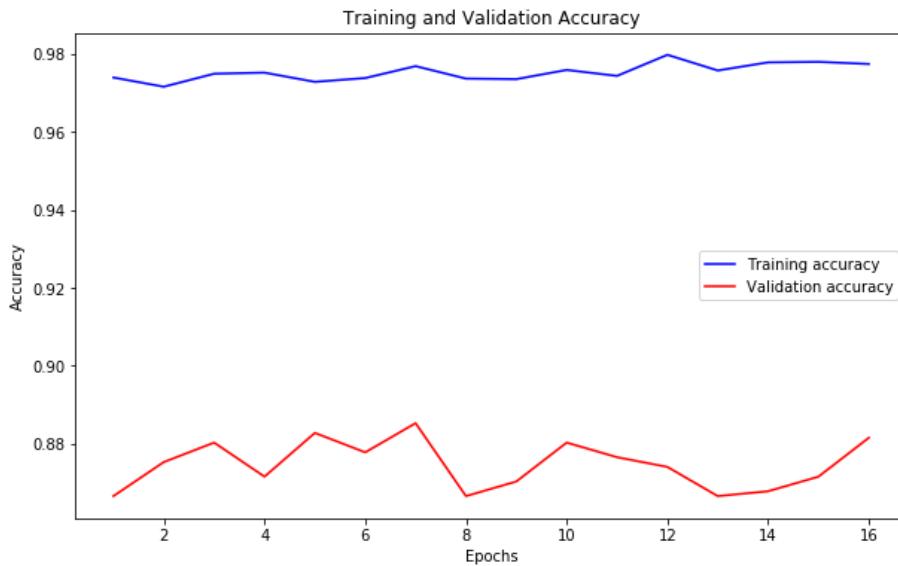


Figure 5.3: Training and validation Accuracy



Figure 5.4: Training and validation Loss

Here the confusion matrices for 15 classes are given for truth value:

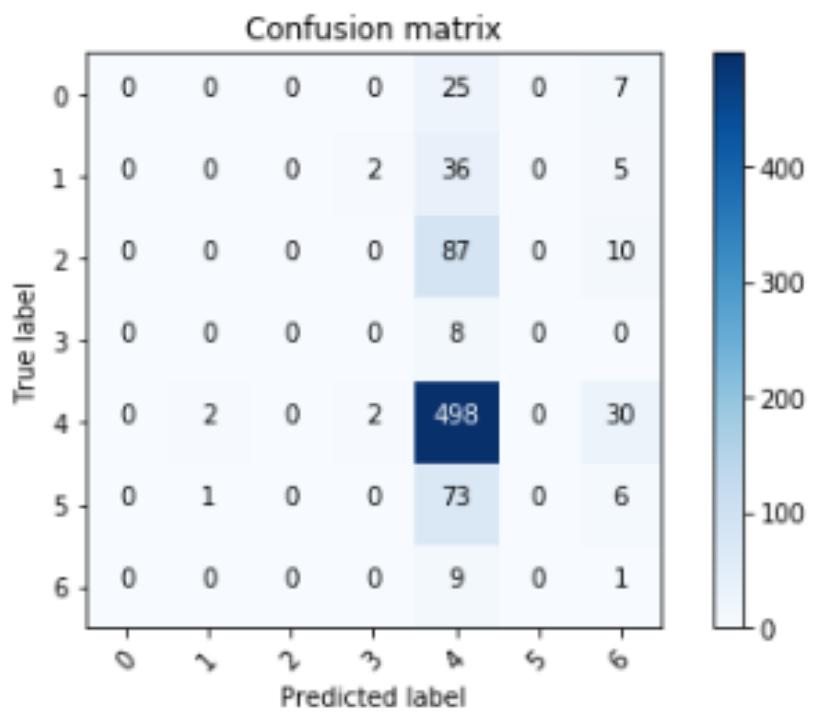


Figure 5.5: confusion matrix 1

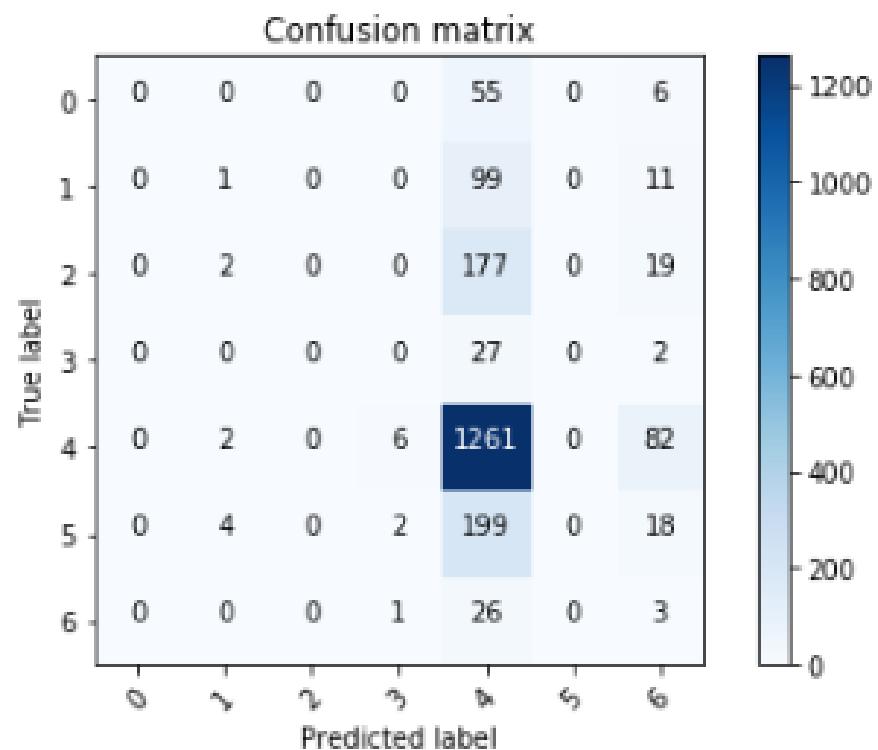


Figure 5.6: confusion matrix 2

Classification Report of this result:

Below a table showing precision, Recall and F1-score and support is given:

	precision	recall	f1-score	support
akiec	0.00	0.00	0.00	61
bcc	0.00	0.00	0.00	111
bkl	0.00	0.00	0.00	198
df	0.00	0.00	0.00	29
mel	0.69	0.92	0.79	1351
nv	0.00	0.00	0.00	223
vasc	0.02	0.07	0.03	30
micro avg	0.64	0.62	0.63	2003
macro avg	0.10	0.14	0.12	2003
weighted avg	0.46	0.62	0.53	2003
samples avg	0.62	0.62	0.62	2003

Figure 5.7: classification report

5.6 Conclusion

On the HAM10000 dataset, MobileNetV2 with LSTM models for skin disease classification was evaluated. The results showed promise with an accuracy of 88.02% and a loss of 65.54%. Despite the model's good performance, overfitting issues were observed, which might mean that the training and validation data sets were different. The model's performance in various classes was provided with useful information by the classification report and confusion matrix. Reducing model complexity or using regularization techniques to combat overfitting could improve generalization performance and assure more robust and reliable skin disease categorization in real-world applications.

Chapter 6

Conclusion and Discussion

6.1 Introduction

A brief summary of the complete research project is provided in this chapter, which includes information on the problem domain, prior research, original contributions, experimental analysis, and conclusions. Furthermore, a brief overview of prospective future research directions is given.

6.2 Conclusion

The accuracy of earlier research on this topic was 85.34%. However, our research proposes novel approaches, including feature reduction, downsizing, normalization, data augmentation, and hyperparameter tuning, that lead to an improved accuracy of 87%. Using these methods, the proposed methodology achieves an enhanced accuracy of 87%, which is higher than previous research findings. The recommended model, which is based on the MobileNet V2 and LSTM method, worked well for the classification and detection of skin illnesses with the least amount of computational resources and effort. The outcome is promising, with an accuracy of 85.34% when tested and compared to alternative methods using real-time pictures from Kaggle. This computationally efficient model would yield better predictions if it combined the LSTM module with the

6.3 Discussion

The suggested method extracts features from skin lesion photos using MobileNet V2, a lightweight convolutional neural network architecture appropriate for mobile and embedded devices. Due to its efficiency, MobileNet V2 is a great choice for real-time applications. Because of its minimal processing resources, it is especially attractive for implementation in clinical settings. Additionally, to capture temporal dependencies in sequential data, the authors use recurrent neural networks (RNNs) in the form of Long Short-Term Memory (LSTM) networks. The sequential nature of dermatological images can be effectively modeled by LSTM in the context of skin disease categorization, capturing patterns and temporal changes that may be suggestive of particular diseases or conditions. The model achieves great accuracy in categorizing different skin disorders by using LSTM for sequence modeling and MobileNet V2 for feature extraction. A thorough assessment of the model's performance is provided by the comprehensive evaluation, which covers measures like accuracy, precision, recall, and F1-score. Nonetheless, there are a few restrictions and areas that need more study. First, there's a chance that the dataset used for testing and training contains intrinsic biases or constraints that could affect how broadly applicable the model is. One of the biggest challenges in this subject is still addressing data paucity and class imbalance in dermatological datasets. Furthermore, even though the suggested model performs well, the complexity of deep learning architectures may limit how easily it can be interpreted. Increasing the interpretability of the model and offering information about the decision-making process may help build healthcare practitioners' confidence and acceptance.

6.4 Limitation

Skin disease datasets frequently exhibit class imbalance, with much fewer samples for some disease categories than for others. As a result, the model may perform biasedly, giving preference to majority classes and underperforming on minority classes. Deep learning models, such as LSTM networks and MobileNet V2, are sometimes referred to as "black-box" models since it can be difficult to understand how they make their judgments. The suggested model may be less helpful in clinical contexts where it is crucial to comprehend the logic underlying forecasts if it cannot be easily interpreted. For real-world applications, deep learning models must be able to generalize well to unknown input. Applying the suggested model to a novel, unidentified skin condition may limit its efficacy due to overfitting to the training data or a lack

of generalization

6.5 Future Work

For improved accuracy, the suggested MobileNet V2 using the LSTM model requires a larger number of parameters. There is not enough significant randomness in the evaluated input image or the outputs from the MobileNet V2 with LSTM model to investigate every potential pattern. In addition to the residual connection bottleneck in the suggested architecture, the model produces better accuracy with less work. By including the self-learning ability and information gained from its prior experiences, the model can be further enhanced. There can be a significant reduction in the model's training effort. To evaluate the significance of the features gathered for every method, the model must be automated, and randomizing elements must be included.

REFERENCES

- [1] M. W. Greaves, “skin disease.” <https://www.britannica.com/science/human-skin-disease>.
- [2] P. N. Srinivasu, J. G. SivaSai, M. F. Ijaz, A. K. Bhoi, W. Kim, and J. J. Kang, “Classification of skin disease using deep learning neural networks with mobilenet v2 and lstm,” *Sensors*, vol. 21, no. 8, p. 2852, 2021.
- [3] I. G. Ferreira, M. B. Weber, and R. R. Bonamigo, “History of dermatology: the study of skin diseases over the centuries,” *Anais Brasileiros de Dermatologia*, vol. 96, pp. 332–345, 2021.
- [4] J. Hunter, “Turning points in dermatology during the 20th century,” *British Journal of Dermatology*, vol. 143, no. 1, pp. 30–40, 2000.
- [5] A. Memon, P. Bannister, I. Rogers, J. Sundin, B. Al-Ayadhy, P. W. James, and R. J. McNally, “Changing epidemiology and age-specific incidence of cutaneous malignant melanoma in england: An analysis of the national cancer registration data by age, gender and anatomical site, 1981–2018,” *The Lancet Regional Health–Europe*, vol. 2, 2021.
- [6] P. K. Sethy, S. K. Behera, and N. Kannan, “Categorization of common pigmented skin lesions (cpsl) using multi-deep features and support vector machine,” *Journal of Digital Imaging*, vol. 35, no. 5, pp. 1207–1216, 2022.
- [7] P. Andrade, M. M. Brites, R. Vieira, A. Mariano, J. P. Reis, O. Tellechea, and A. Figueiredo, “Epidemiology of basal cell carcinomas and squamous cell carcinomas in a department of dermatology: a 5 year review,” *Anais brasileiros de dermatologia*, vol. 87, pp. 212–219, 2012.
- [8] A. Laga, “Surgical pathology of melanocytic neoplasms,” in *Pathobiology of Human Disease* (L. M. McManus and R. N. Mitchell, eds.), pp. 3563–3591, San Diego: Academic Press, 2014.

- [9] M. Meleti, W. J. Mooi, M. K. Casparie, and I. van der Waal, “Melanocytic nevi of the oral mucosa—no evidence of increased risk for oral malignant melanoma: An analysis of 119 cases,” *Oral oncology*, vol. 43, no. 10, pp. 976–981, 2007.
- [10] C. Hafner and T. Vogt, “Seborrheic keratosis,” *JDDG: Journal der Deutschen Dermatologischen Gesellschaft*, vol. 6, no. 8, pp. 664–677, 2008.
- [11] P. R. Cohen, C. P. Erickson, and A. Calame, “Atrophic dermatofibroma: a comprehensive literature review,” *Dermatology and Therapy*, vol. 9, pp. 449–468, 2019.
- [12] S. Astner and R. R. Anderson, “Treating vascular lesions,” *Dermatologic therapy*, vol. 18, no. 3, pp. 267–281, 2005.
- [13] J. V. Schmitt and H. A. Miot, “Actinic keratosis: a clinical and epidemiological revision,” *Anais brasileiros de dermatologia*, vol. 87, pp. 425–434, 2012.
- [14] G. Guorgis, C. D. Anderson, L. Johan, and F. Magnus, “Actinic keratosis diagnosis and increased risk of developing skin cancer: a 10-year cohort study of 17,651 patients in sweden,” *Acta Dermato-Venereologica*, vol. 100, no. 8, 2020.
- [15] J. P. Callen, D. R. Bickers, and R. L. Moy, “Actinic keratoses,” *Journal of the American Academy of Dermatology*, vol. 36, no. 4, pp. 650–653, 1997.
- [16] C. J. Cockerell, “Histopathology of incipient intraepidermal squamous cell carcinoma (“actinic keratosis”),” *Journal of the American Academy of Dermatology*, vol. 42, no. 1, pp. S11–S17, 2000.
- [17] C. Wong, R. Strange, and J. Lear, “Basal cell carcinoma,” *Bmj*, vol. 327, no. 7418, pp. 794–798, 2003.
- [18] D. Schadendorf, A. C. Van Akkooi, C. Berking, K. G. Griewank, R. Gutzmer, A. Hauschild, A. Stang, A. Roesch, and S. Ugurel, “Melanoma,” *The Lancet*, vol. 392, no. 10151, pp. 971–984, 2018.
- [19] H. S. Black, “Skin cancer,” *Heber, D., Blackburn, GL, Go, VLW, Eds*, pp. 405–422, 2006.
- [20] 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, no. 7639, pp. 115–118, 2017.

- [21] M. Gurucharan, “Basic cnn architecture: Explaining 5 layers of convolutional neural network,” *URL: https://www.upgrad.com/blog/basic-cnn-architecture*, 2020.
- [22] P. Ratan, “„what is the convolutional neural network architecture?“,” *Analytics Vidhya*, vol. 28, 2020.
- [23] Y.-C. Chen, D. J.-K. Hong, C.-W. Wu, and M. Mupparapu, “The use of deep convolutional neural networks in biomedical imaging: A review,” *Journal of Orofacial Sciences*, vol. 11, no. 1, pp. 3–10, 2019.
- [24] B. H. Nayef, S. N. H. S. Abdullah, R. Sulaiman, and Z. A. A. Alyasseri, “Optimized leaky relu for handwritten arabic character recognition using convolution neural networks,” *Multimedia Tools and Applications*, pp. 1–30, 2022.
- [25] J.-H. Lee, D.-h. Kim, S.-N. Jeong, and S.-H. Choi, “Diagnosis and prediction of periodontally compromised teeth using a deep learning-based convolutional neural network algorithm,” *Journal of periodontal & implant science*, vol. 48, no. 2, p. 114, 2018.
- [26] L. Zhang and H. Schaeffer, “Forward stability of resnet and its variants,” *Journal of Mathematical Imaging and Vision*, vol. 62, pp. 328–351, 2020.
- [27] L. Cai, Z. Wang, R. Kulathinal, S. Kumar, and S. Ji, “Deep low-shot learning for biological image classification and visualization from limited training samples,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 5, pp. 2528–2538, 2021.
- [28] S. Targ, D. Almeida, and K. Lyman, “Resnet in resnet: Generalizing residual architectures,” *arXiv preprint arXiv:1603.08029*, 2016.
- [29] C. D. Baso and A. A. Ramos, “Enhancing sdo/hmi images using deep learning,” *Astronomy & Astrophysics*, vol. 614, p. A5, 2018.
- [30] N. Van Noord and E. Postma, “Learning scale-variant and scale-invariant features for deep image classification,” *Pattern Recognition*, vol. 61, pp. 583–592, 2017.
- [31] N. Dong, L. Zhao, C.-H. Wu, and J.-F. Chang, “Inception v3 based cervical cell classification combined with artificially extracted features,” *Applied Soft Computing*, vol. 93, p. 106311, 2020.

- [32] Y. Fan, J. Li, U. A. Bhatti, C. Shao, C. Gong, J. Cheng, and Y. Chen, “A multi-watermarking algorithm for medical images using inception v3 and dct,” *CMC-Computers Materials & Continua*, vol. 74, no. 1, pp. 1279–1302, 2023.
- [33] M. P. Ayyar, J. Benois-Pineau, and A. Zemmari, “A feature understanding method for explanation of image classification by convolutional neural networks,” in *Explainable Deep Learning AI*, pp. 79–96, Elsevier, 2023.
- [34] L. Ali, F. Alnajjar, H. A. Jassmi, M. Gocho, W. Khan, and M. A. Serhani, “Performance evaluation of deep cnn-based crack detection and localization techniques for concrete structures,” *Sensors*, vol. 21, no. 5, p. 1688, 2021.
- [35] R. Sadik, A. Majumder, A. A. Biswas, B. Ahammad, and M. M. Rahman, “An in-depth analysis of convolutional neural network architectures with transfer learning for skin disease diagnosis,” *Healthcare Analytics*, vol. 3, p. 100143, 2023.
- [36] A. Ben-Hur and J. Weston, “A user’s guide to support vector machines,” *Data mining techniques for the life sciences*, pp. 223–239, 2010.
- [37] K. S. Parikh and T. P. Shah, “Support vector machine—a large margin classifier to diagnose skin illnesses,” *Procedia Technology*, vol. 23, pp. 369–375, 2016.
- [38] O. K. Pal, “Skin disease classification: A comparative analysis of k-nearest neighbors (knn) and random forest algorithm,” in *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*, pp. 1–5, IEEE, 2021.
- [39] S. Jaiswal, “K-nearest neighbor (knn) algorithm for machine learning,” 2015.
- [40] K. Manjusha, K. Sankaranarayanan, and P. Seena, “Prediction of different dermatological conditions using naïve bayesian classification,” *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, no. 1, 2014.
- [41] S. Moldovanu, C.-D. Obreja, K. C. Biswas, and L. Moraru, “Towards accurate diagnosis of skin lesions using feedforward back propagation neural networks,” *Diagnostics*, vol. 11, no. 6, p. 936, 2021.
- [42] O. Imarenezor, “Epidemiology and pathophysiology of common skin diseases in west africa: An immunodermatological framework,” 2020.

- [43] L. Yu, H. Chen, Q. Dou, J. Qin, and P.-A. Heng, “Automated melanoma recognition in dermoscopy images via very deep residual networks,” *IEEE transactions on medical imaging*, vol. 36, no. 4, pp. 994–1004, 2016.
- [44] X. Zhang, S. Wang, J. Liu, and C. Tao, “Towards improving diagnosis of skin diseases by combining deep neural network and human knowledge,” *BMC medical informatics and decision making*, vol. 18, pp. 69–76, 2018.
- [45] C. K. Sen, G. M. Gordillo, S. Roy, R. Kirsner, L. Lambert, T. K. Hunt, F. Gottrup, G. C. Gurtner, and M. T. Longaker, “Human skin wounds: a major and snowballing threat to public health and the economy,” *Wound repair and regeneration*, vol. 17, no. 6, pp. 763–771, 2009.
- [46] V. B. Kumar, S. S. Kumar, and V. Saboo, “Dermatological disease detection using image processing and machine learning,” in *2016 Third International Conference on Artificial Intelligence and Pattern Recognition (AIPR)*, pp. 1–6, IEEE, 2016.
- [47] P. Tschandl, C. Rosendahl, and H. Kittler, “The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions,” *Scientific data*, vol. 5, no. 1, pp. 1–9, 2018.
- [48] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.
- [49] R. C. Staudemeyer and E. R. Morris, “Understanding lstm—a tutorial into long short-term memory recurrent neural networks,” *arXiv preprint arXiv:1909.09586*, 2019.
- [50] K. Says, “What is lstm-introduction to long short term memory,” *Intellipaat Blog*, 2020.
- [51] B. Or, “Deep learning for inertial navigation a short review of cutting edge deep learning-based solutions for inertial navigation.,”
- [52] P. N. Huu, N. N. Thi, and T. P. Ngoc, “Proposing posture recognition system combining mobilenetv2 and lstm for medical surveillance,” *IEEE Access*, vol. 10, pp. 1839–1849, 2021.