T.C. İSTANBUL AREL UNIVERSITY
FACULTY OF ENGINEERING
DEPARTMENT OF INDUSTRIAL ENGINEERING

SKIN DISEASE DETECTION USING THE HAM1000 DATASET AND CONVOLUTIONAL NEURAL NETWORKS

INTERDISCIPLINARY PROJECT

AYŞEGÜL HİLAL SEYHAN
MUHAMMED AKBAR
HALIL IBRAHIM ALTINBAŞ

ISTANBUL,2024

T.C.

İSTANBUL AREL UNIVERSITY

FACULTY OF ENGINEERING AND ARCHITECTURE

DEPARTMENT OF INDUSTRIAL ENGINEERING

Skin Disease Detection Using the HAM1000 Dataset and Convolutional Neural Networks

INTERDISCIPLINARY PROJECT

Ayşegül Hilal SEYHAN
Muhammed AKBAR
Halil İbrahim ALTINBAŞ

Supervisor: Sabahattin Kerem AYTULUN

İSTANBUL, Janurary 2024

T.C.
İSTANBUL AREL UNIVERSITY
FACULTY OF ENGINEERING AND ARCHITECTURE
DEPARTMENT OF INDUSTRIAL ENGINEERING

Name of the project: Skin Disease Detection Using the HAM1000 Dataset and Convolutional Neural Networks

Name/Last Name of the Student: Ayşegül Hilal SEYHAN, Muhammad AKBAR, Halil Ibrahim ALTINBAŞ

DATE:

SABAHATTİN KEREM AYTULUN

We hereby state that we have held the graduation examination of *name of the student* and agree that the student has satisfied all requirements.

THE EXAMINATION COMMITTEE

Committee Member	Signature
1. …….	……...
2. ……..……...	……...
3. ……..……...	…….

ABSTRACT

SKIN DISEASE DETECTION USING THE HAM1000 DATASET AND CONVOLUTIONAL NEURAL NETWORKS

Ayşegül Hilal SEYHAN
Muhammad AKBAR
Halil Ibrahim ALTINBAŞ
Interdisciplinary Project, Department of Industrial Engineering
Advisor: Sabahattin Kerem AYTULUN
2024

ABSTRACT

The skin, a vital organ that serves as a protective shield for the human body, is susceptible to various infections caused by fungi, viruses, and even environmental factors like dust. Conditions such as acne and eczema can cause significant physical and emotional distress. Proper diagnosis is crucial for effective treatment, alleviating the suffering of those affected. This study aims to develop a robust system for the automated detection and classification of skin diseases using Convolutional Neural Networks (CNNs), a specialized type of neural network well-suited for image analysis.

The proposed system employs advanced image processing techniques, where users can upload photos of affected skin areas for analysis. These images are processed on a central server, which uses CNN models to classify the condition and provide a diagnosis. The application focuses on enhancing diagnostic precision and efficiency, addressing the growing demand for automated solutions to assist medical practitioners in expediting treatment plans.

Skin diseases are a global health concern, often posing significant risks that can lead to severe physical and mental health consequences, including skin cancer. Diagnosing these conditions from clinical images is inherently challenging due to the complexity and variability of skin lesions. Manual diagnosis, while effective, can be subjective and time-consuming, further emphasizing the need for automated systems.

This study integrates deep learning with collective intelligence to develop an advanced skin lesion classification framework. Using multiple CNNs trained on the HAM10000 dataset—a benchmark dataset containing diverse dermatoscopic images—this system identifies seven types of skin lesions, including melanoma. By leveraging the hierarchical learning capabilities of CNNs, the framework ensures high accuracy in classifying skin diseases, thereby contributing to early detection, efficient treatment, and improved outcomes for patients worldwide.

Keywords: Skin Disease Classification, CNN’s , Deep Learning, Skin lesions

ACKNOWLEDGEMENTS

We, Ayşegül Hilal SEYHAN, Muhammed AKBAR, and Halil İbrahim ALTINBAŞ, would like to extend our heartfelt thanks to all those who have played a part in the successful completion of our project.

Our deepest gratitude goes to our distinguished mentor, Professor Sabahattin Kerem AYTULUN and Pınar KARADAYI ATAŞ. Their expert guidance, invaluable advice, and constant encouragement have been instrumental to the progress and success of this project. We are deeply appreciative of their unwavering support, willingness to always make time for our inquiries, and their thoughtful feedback. Their dedication to our growth and open-door approach have been a true source of inspiration throughout this journey.

We are also grateful to our professors, peers, and colleagues who shared their knowledge and perspectives with us along the way. Their constructive feedback, thoughtful suggestions, and words of encouragement have been essential in shaping the final outcome of this project.

Finally, we wish to acknowledge everyone who contributed to our success, whether directly or indirectly. Your support, both seen and unseen, has been vital to the completion of this work.

Thank you all for being an integral part of our journey. Your contributions have made all the difference.

TABLE OF CONTENTS

LIST OF FIGURE

LIST OF ABBREVIATIONS

1. INTRODUCTION

Skin diseases, ranging from relatively benign conditions to some of the most severe and life-threatening ailments, are among the most prevalent health problems globally, affecting millions of individuals. Early and accurate diagnosis of skin diseases is critical, as timely intervention often determines the success of treatment and patient outcomes. Traditional diagnostic methods, while informative, demand significant expertise and time, creating challenges in resource-constrained settings. This highlights the need for innovative, automated solutions to enhance diagnostic accuracy and efficiency.

The World Health Organization's First Global Meeting on Skin-Related Neglected Tropical Diseases (skin NTDs) in March 2023 in Geneva, Switzerland, underscored the widespread impact of skin conditions, estimating that approximately 1.1 billion people are affected at any given time. Skin infections—whether bacterial, viral, fungal, or parasitic—are among the most frequent diseases in tropical and resource-poor regions. In many communities, skin NTDs constitute around 10% of all skin diseases. These findings emphasize the importance of adopting community-based strategies in endemic regions to address skin NTDs and other skin ailments comprehensively as part of Universal Health Coverage (UHC) and the Sustainable Development Goals (SDGs), ensuring no one is left behind.

Convolutional Neural Networks (CNNs) have emerged as a promising tool in automating skin disease diagnosis over the past decade. Their ability to hierarchically learn features from medical images enables the identification of complex patterns in skin lesions, aiding early detection and classification of diseases. Accurate segmentation and classification of skin lesions are crucial for identifying potential malignancies, particularly in cases of melanoma, one of the deadliest forms of skin cancer. Early detection significantly enhances treatment success rates, while delayed diagnosis increases the risk of metastasis and can lead to severe outcomes, including death.

Skin cancers are categorized into four primary types: basal cell carcinoma, squamous cell carcinoma, Merkel cell carcinoma, and melanoma. Among these, melanoma is the most aggressive, accounting for the highest mortality rate. Despite its relatively low overall mortality rate (approximately 10%), the treatment process for skin cancer is often painful, and late-stage diagnosis can result in severe consequences, including amputation of the affected area. The urgency of early diagnosis is underscored by projections indicating a significant rise in cancer rates in the coming years—24.1% in men and 20.6% in women.(Popescu , El-Khati , Ichim , 2022)

Automated diagnostic systems, particularly those leveraging neural network-based architectures, are becoming indispensable in dermatology. A systematic review of neural network-based systems for melanoma detection has highlighted the growing role of artificial intelligence in improving diagnostic efficiency and accuracy. The HAM10000 dataset, a benchmark resource comprising 10,000 dermatoscopic images representing diverse skin conditions, plays a pivotal role in training and evaluating machine learning models for skin lesion classification. (Akram , Lodhi , Naeem , Alhaisoni , Ali , Haider , Qadri , 2020)

Despite significant advancements, challenges remain in developing robust systems for classifying multiple skin disease categories. Previous research has often focused on a single disease or demonstrated limitations in handling the nuanced similarities between different skin conditions. This study aims to address these gaps by leveraging the HAM10000 dataset to develop a CNN model capable of accurately classifying multiple skin lesion categories. By contributing to the field of dermatology and enhancing diagnostic precision, this research aspires to advance early detection and treatment strategies for skin diseases worldwide. (Al-masni , Kim , Kim , 2020)

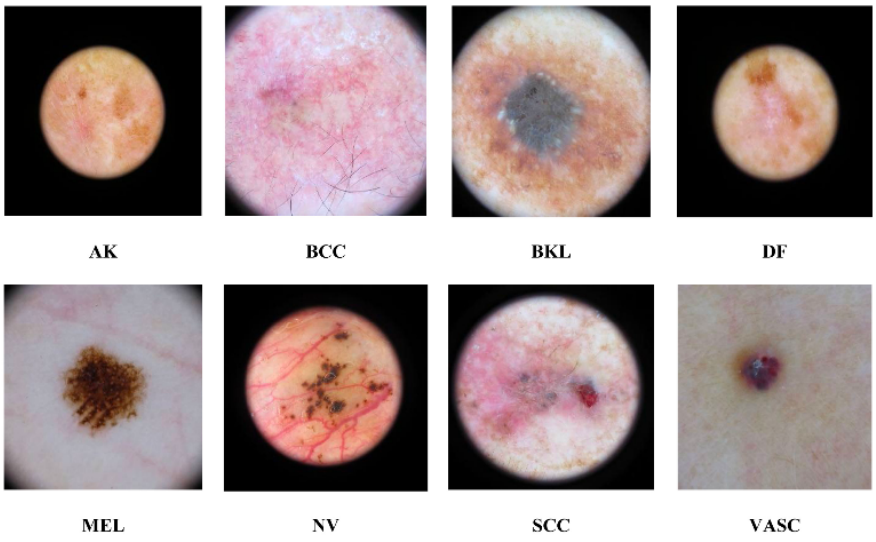

Figure 1: Example of skin lesions used for classification.

1.1 Purpose of the Study

The purpose of this study is to develop and evaluate a Convolutional Neural Network (CNN) model capable of accurately classifying multiple skin lesion categories using the HAM10000 dataset. By addressing existing limitations in automated diagnostic systems such as their focus on single diseases or challenges in distinguishing nuanced similarities between skin condition this study seeks to advance the field of dermatology through the application of innovative deep learning techniques. Ultimately, this research aims to enhance early detection and diagnostic precision for a broad range of skin diseases, improving treatment outcomes and contributing to global health initiatives like Universal Health Coverage (UHC) and the Sustainable Development Goals (SDGs). The proposed model aspires to provide a robust and scalable solution, particularly for resource-constrained and endemic regions, where early and accurate diagnosis is critical for effective healthcare delivery.

2. RELATED WORKS

The human skin, a remarkable organ, frequently encounters both common and rare diseases, making skin disease diagnosis one of the most ambiguous and challenging areas of study. A significant number of cases remain unreported due to insufficient medical infrastructure and support. To address this, various artificial intelligence (AI) techniques have been employed, starting with handcrafted feature extraction followed by classifier training. AI has shown potential in identifying and assessing skin conditions such as melanoma, basal cell carcinoma, and psoriasis.

A prominent approach involves convolutional neural networks (CNNs), which excel at detecting the intricate features necessary for accurate pattern recognition. For instance, CNNs have been used to distinguish clinical images of rosacea patients from individuals with acne, seborrheic dermatitis, and eczema. In one study, a hybrid method was proposed, combining saliency segmentation using an enhanced high-dimension contrast transform (HDCT) with binary images generated by a 16-layer CNN model. The classification module utilized transfer learning to retrain a DenseNet201 model on segmented lesion images, achieving high accuracy in skin lesion classification.

Recent advancements in medical imaging segmentation and classification further highlight the role of deep learning. Amine Ben Slama et al. developed an improved segmentation method for tumor detection in MRI images using ResNet architecture and VGG gliomas grading. This method achieved better performance than the BRATS 2020 for brain tumor image segmentation, emphasizing the importance of automated segmentation for accurate tumor size measurement.

Deepak Jayaprakash Doggalli and Sunil Kumar B S assessed the U-Net model's efficacy in segmenting liver tumors from abdominal CT images, achieving a dice global score of 94% for liver segmentation and 73% for tumor segmentation on the LiTS dataset. Similarly, Saeed Iqbal and Adnan N. Qureshi introduced BreastUNet, a novel CNN model designed for mitotic nuclei evaluation in breast histopathology images. This model achieved an F1 score of 95% and demonstrated its value as a supplementary tool for pathologists.

Other notable contributions include Aarthi Chelladurai et al.'s automated model for early Alzheimer’s diagnosis using functional MRI images. Their method incorporated image normalization, segmentation, feature extraction via Gabor wavelets and Gray Level Co-Occurrence Matrix (GLCM), feature reduction with the Honey Badger Optimization Algorithm (HBOA), and classification using a Multi-Layer Perceptron (MLP). This approach achieved a classification accuracy of 99.44%, outperforming traditional methods.

For brain malignancies, Zahid Rasheed et al. applied methodologies yielding superior accuracy compared to pre-trained models like VGG16, ResNet50, and InceptionV3. Their approach achieved an overall classification accuracy of 97.84%, emphasizing the potential for extension to subtype classification of brain tumors.

Moreover, the effectiveness of traditional algorithms such as Naïve Bayes (NB) and K-Nearest Neighbor (K-NN) has been improved using Bayesian boost and bagging techniques. K-NN showed a performance increase of over 7% compared to NB. Similarly, model

stacking methodologies have demonstrated a 25% improvement in accuracy for neural networks, achieving a peak accuracy of 95.66%. These findings highlight the importance of data preprocessing and addressing class imbalance in enhancing classification performance.

2.1 Medical Image Classification

The application of machine learning in medical image classification has gained traction in recent years. Early approaches primarily utilized traditional machine learning algorithms, which required extensive feature engineering and domain expertise. However, these methods often struggled with generalization to unseen classes, particularly in the context of rare skin diseases.

With the advent of deep learning, CNNs have emerged as a powerful tool for image classification tasks. CNNs automatically learn hierarchical features from raw pixel data, eliminating the need for manual feature extraction. Numerous studies have demonstrated the effectiveness of CNNs in various medical imaging tasks, including skin disease detection. For instance, Esteva et al. (2017) showcased a deep learning model that achieved performance comparable to dermatologists in melanoma classification.

3. METHODOLOGY

The HAM1000 dataset is a publicly available collection of dermatoscopic images, curated to facilitate research in skin lesion classification. It includes images of various skin conditions, such as melanoma, nevus, and basal cell carcinoma, along with associated metadata. The dataset is divided into training, validation, and test sets, allowing for robust model evaluation. The diversity of the dataset makes it a valuable resource for developing and benchmarking machine learning models in dermatology.

3.1 Dataset Overview

The HAM10000 dataset is a publicly available collection of 10,000 dermatoscopic images curated to facilitate research in skin lesion classification. It includes images labeled with one of seven skin lesion categories: melanoma, nevus, basal cell carcinoma, actinic keratosis, dermatofibroma, vascular lesions, and seborrheic keratosis. The dataset is accompanied by metadata and is divided into training, validation, and test sets, enabling robust evaluation of machine learning models.

The diversity in image size, resolution, and quality reflects real-world challenges, making the dataset a valuable resource for developing and benchmarking machine learning algorithms in dermatology. To ensure uniformity and optimize model performance, preprocessing steps such as resizing, normalization, and augmentation are often necessary. By providing a

comprehensive representation of various skin conditions, HAM10000 plays a pivotal role in advancing research in skin lesion detection and classification.

3.2 Existing Methods

Recent research in the field of skin lesion detection has shown a growing trend toward using a combination of various techniques and deep learning models to achieve higher accuracy and better generalization. Many of these systems leverage multiple classifiers and deep convolutional neural networks (CNNs) in different architectures to tackle the challenges posed by skin lesion classification.

3.3 Combination of ABCDE with Support Vector Machine (SVM)

Some studies have explored combining traditional techniques such as the ABCDE rule (Asymmetry, Border, Color, Diameter, and Evolution) with machine learning classifiers like SVM to improve classifi- cation accuracy. This hybrid approach often capitalizes on the robustness of traditional heuristics and the power of machine learning models.

3.4 Transfer Learning and Modified Networks

Another common strategy is the use of modified neural networks, particularly with transfer learning. This method involves fine-tuning pre-trained models on skin lesion datasets to adapt them to the specific task of lesion detection. By leveraging the knowledge embedded in large models, researchers have been able to achieve promising results in skin lesion classification .(Popescu, El-Khatib, Ichim , 2022)

3.5 Multiple Networks and Ensemble Models

Multiple networks are also employed in an ensemble fashion, where each model contributes a part to the overall system’s decision. For example, one network may perform lesion segmentation while another takes over the classification task using the segmented data. This approach is particularly effective in dealing with complex datasets where different aspects of the image (e.g., color, shape, texture) need to be considered for accurate diagnosis. Ensemble models like these have been found to outperform single networks in terms of accuracy and robustness.(Esteva , Kuprel , Novoa , Ko , Swetter , Blau, Thrun , 2017)

3.6 Pre-trained Networks for Feature Extraction

Convolutional neural networks (CNNs) such as AlexNet, ResNet, and GoogLeNet have been widely adopted for their excellent feature extraction capabilities. These models are trained on large datasets and fine-tuned for skin lesion classification. For instance, Esteva

et al. 2017 used GoogLeNet Inception- V3 to achieve dermatologist-level classification of skin cancer, showing its effectiveness in the domain. AlexNet, though older, has still shown strong performance, achieving an accuracy of 93.64% in some skin mole detection systems.(Pomponiu , 2016)

3.7 Residual Networks (ResNet) and Deep Learning

As CNNs have become deeper, the issue of vanishing gradients has prompted the use of residual networks like ResNet-50 and ResNet-101. These architectures help improve the accuracy of deep models by allowing gradients to flow more effectively through the network. Studies have demonstrated the use of ResNet-50 in melanoma classification, with accuracy reaching up to 90.67% when combined with handcrafted features Similar success has been seen with networks like Xception and MobileNet- V2, which are also frequently applied in skin lesion diagnosis due to their lightweight and efficient architecture. (lmaraz-Damian , 2020)

3.8 DenseNet and Feature Fusion

DenseNet-201 is another frequently used architecture, especially for feature extraction and classification. Research has combined DenseNet-201 with fully convolutional networks for segmentation, achieving up to 81.29% accuracy on the ISIC 2017 dataset . This model benefits from its densely connected layers, which improve feature reuse and facilitate learning. (Al-masni , 2020)

3.9 Hybrid Models for Skin Disease Classification

Hybrid models that combine different architectures and classifiers have shown promise in tackling a range of skin diseases. For example, a multi-class system combining ResNet, AlexNet, and other CNN models has been proposed to classify various types of skin lesions, including melanoma, using ensemble voting systems to enhance prediction accuracy.(Ichim, 2020)

3.10 Decision Fusion and Voting Systems

Researchers have also developed complex voting systems that combine the outputs of several mod- els. These systems typically rely on the outputs from multiple CNNs trained for different tasks (e.g., melanoma vs. non-melanoma classification) and aggregate the results using a decision-making process like majority voting or confidence-weighted averaging. One such example is the multi-network voting system proposed by Gong et al. 2020, which combines the strengths of various individual networks for accurate lesion classification.

In conclusion, the trend in skin lesion classification is moving towards ensemble models, transfer learning, and multi-network systems that aim to leverage the strengths of multiple techniques to achieve higher accuracy and robustness. This approach is expected to remain a significant direction in future research, particularly with the continuous evolution of neural network architectures and the availability of larger and more diverse datasets.

4. MATERIALS

Dataset Used There are many datasets with skin lesions: PH2, ISIC 2016, ISIC 2017, ISIC 2018- HAM10000, ISIC 2019, ISIC 2020, DERMQUEST, MED-NODE, DERMNET, DERMIS, DERMOFIT, etc. [3]. HAM10000 is one of the largest skin lesions datasets publicly available for academic research. In this paper, for current experiments, we chose to use the HAM10000 ("Human Against Machine with 10,000 training images") dataset, introduced in the ISIC 2018 challenge, which contains 10,015 dermatoscopic images which can serve as a training dataset for academic machine learning purposes. (Al-masni , 2020) The HAM10000 dataset covers image samples for all-important diagnostic categories (classes) in the real pigmented lesions:

- Actinic keratoses and intraepithelial carcinoma/Bowen's disease (akiec)
- Basal cell carcinoma (bcc)
- Benign keratosis-like lesions (bkl-solar lentigines/seborrheic keratoses and lichen-planus like keratoses)
- Dermatofibroma (df)
- Melanoma (mel)
- Melanocytic nevi (nv)
- Vascular lesions (vasc-angiomas, angiokeratomas, pyogenic granulomas, and hemorrhage).

contains 1015 JPEG images and is split into two packages/folders: HAM10000 imagespart1

(5000 JPEG images) and HAM10000 imagespart2 (5015 JPEG images). Some examples are presented below in Figure 2.

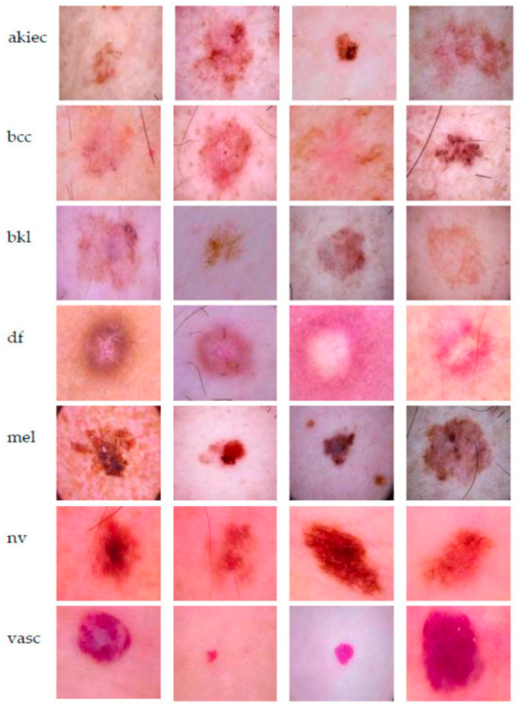

Figure 2: Example of each class from HAM10000 dataset

4.1 Data Preprocessing

Before training the CNN model, we perform several preprocessing steps:

Image Resizing: All images are resized to different uniform dimension (e.g., 224x224, 56,56 pixels) to ensure compatibility with different types of models in CNN architecture.

Normalization: Pixel values are normalized to a range of [0, 1] to improve convergence during training. Data Augmentation: To enhance the model's robustness and prevent overfitting, we apply data

augmentation techniques, including random rotations, flips, and brightness adjustments. This increases the diversity of the training dataset without the need for additional labeled data. Data augmentation can be useful in the training phase if the data in certain classes of the dataset is small. This can reduce the overfitting of the deep neural networks. all of this occurs in the part

We used dataset augmentation methods to try to balance the data (classes). As can be seen in Figure 3, the following augmentation methods were applied:

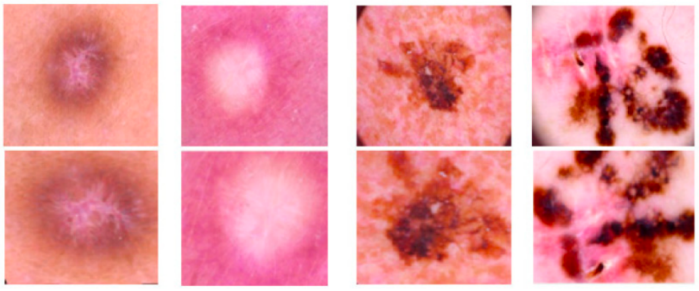

Figure 3: Vertical or horizontal pixel shift with a maximum of 10

5. CNN ARCHITECTURE

Convolutional Neural Networks (CNNs) are very good in processing data such as pictures, speech, and audio, frequently outperforming other neural network models. Their structure is based on three types of layers: convolutional, pooling, and fully connected (FC). A CNN relies on important components such as input data, filters, and feature maps, with these layers handling the majority of the processing.
Pooling layers, also known as down-sampling layers, are responsible for lowering the dimensionality of input data by limiting the number of parameters.

Pooling layers, like convolutional layers, apply a filter to the input data; however, these filters are not weighted. Pooling is classified into two types: maximum pooling, which picks the greatest value from an area, and average pooling, which calculates the average value. The network's ultimate component is the fully connected layer (FC). Unlike partially connected layers, which do not directly link all pixel values to the output, each node in a fully connected layer links to all nodes in the preceding layer, allowing spatial information to be integrated into the final output.

CNN Model Architecture
The CNN model is built using a series of repeated layers, as detailed below:

- **Convolutional Layer:** This is the basic layer of a CNN, where feature extraction takes place. The raw data is filtered to find patterns under particular conditions. Neurons in this layer are organized in three dimensions, allowing the network to record spatial and hierarchical characteristics.
- **Max-Pooling Layer:** Pooling layers are added after each convolutional layer to assist minimize the size of the feature maps, thereby compressing information while retaining critical features. These layers utilize tiny rectangular filters to extract values from convolutional output blocks. The most frequent pooling approach is max pooling, which retains the maximum pixel value from each block.

- **Fully Connected Layer:** The fully connected layer is the CNN's last layer, and it connects all neurons from the previous levels to the output layer. This layer collects

learnt characteristics and lowers spatial input in a manner similar to a typical artificial neural network. It starts with input neurons and ends with output neurons, which enable the ultimate classification or prediction job.

In this study, we explore the use of Convolutional Neural Networks (CNNs) for skin disease detection using the HAM10000 dataset. We evaluate and compare two different approaches to build the model architecture: a single CNN model and different ready made image classification models from keras. Each approach has distinct advantages, and their implementation is discussed below.

5.1 Standard CNN Model

The standard CNN model is a single deep learning network that is typically trained from scratch or fine-tuned using a pre-trained model on a large dataset like ImageNet. The goal of the CNN model is to automatically extract features from input images through layers of convolutions, pooling, and fully connected layers to classify the data.

5.1.1 Architecture of the Standard CNN Model

The architecture of the standard CNN model follows a typical design pattern:

• **Input Layer:** The input to the model is an image with a fixed size, typically 224x224 or 256x256 pixels, and 3 color channels (RGB) but due to limited resources i have used 58x58 pixel images in my CNN model.

• **Convolutional Layers:** The convolutional layers apply various filters to the input images, extract- ing hierarchical features like edges, textures, and shapes. These layers are followed by activation functions (e.g., ReLU) to introduce non-linearity.

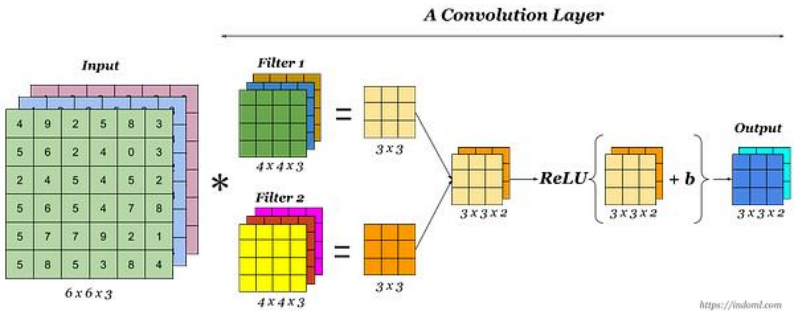

Figure 4: Illustration of convolutional layers extracting features

• **Pooling Layers:** After the convolutional layers, pooling layers (usually MaxPooling) reduce the spatial dimensions of the feature maps, making the model more computationally efficient while preserving important information.

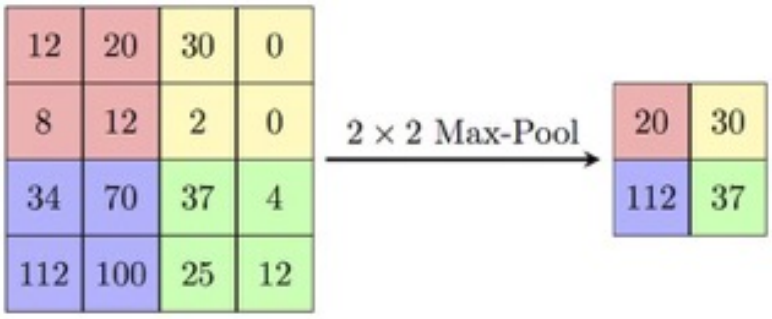

Figure 5: Illustration of a functioning Max Pooling layer

• **Fully Connected Layers:** After the feature extraction layers, the output is flattened and passed through fully connected (dense) layers. These layers are used to combine the features learned from the previous layers and make the final prediction.

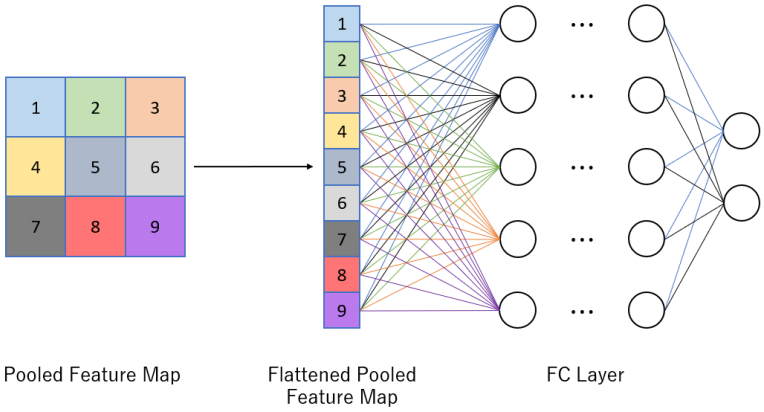

Figure 6: Fully Connected Layer

• **Output Layer:** The output layer uses a softmax activation function for multi-class classification. Each class corresponds to a specific skin disease type, and the network outputs a probability distribution for each class.

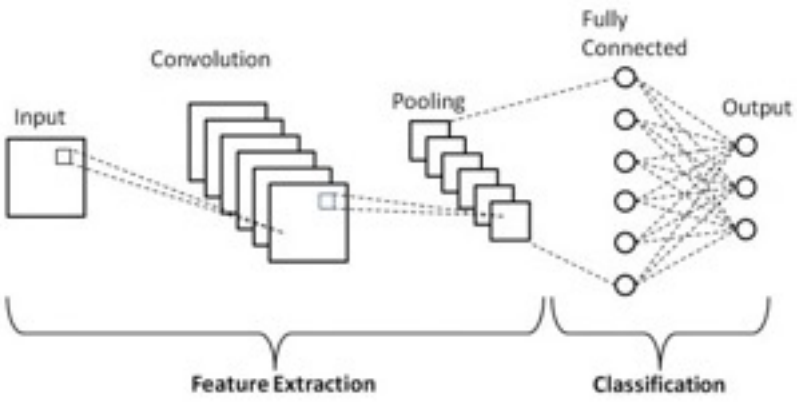

Figure 7: All Layers performing to give results that contribute in predicting the class using output layer

5.1.2 Advantages of the Standard CNN Model

The standard CNN model has several advantages:

- It is straightforward to implement and train on a given dataset.
- Automatic feature extraction.
- Highly accurate at image recognition classification. (although our model may have some tendency to not show 100% performance but that is absolutely not because the structure of cnn lacks ability or the algorithm is faulty)
- It requires fewer resources compared to more complex architectures like ensemble models.
- their ability to perform automatic feature extraction or feature learning.

  However, the model's performance is limited by its single architecture, which may not capture all the complex features of the dataset, leading to overfitting or reduced accuracy on unseen data.

5.2 Famous CNN Models

Apart from this CNN model we are building from scratch, we will make use of and analyse different ready made CNN models on our HAM1000 dataset to discover abilities and weaknesses of different algorithms

5.2.1 Architecture of different models

For the sake of our project, we shall perform detailed analysis of impact of different ready made models on several pre-trained CNN models, such as ResNet50, VGG16, and DenseNet, to form a stronger classifier. Each model is trained independently, and their predictions are combined to improve accuracy.

5.3 Detailed Architecture of CNN Models

5.3.1 VGG16

VGG16 is a deep convolutional neural network that has 16 layers with weights. It follows a simple and uniform architecture, using only 3 × 3 convolutional filters and 2 × 2 max-pooling layers.

- Input Layer: Accepts an input image of size 224 × 224 × 3 (RGB image).
- Convolutional Layers:
  - – Consists of 13 convolutional layers.

- – All convolutional layers use 3 × 3 filters with a stride of 1 and padding to preserve the spatial

  resolution.

- Pooling Layers:
  – 5 max-pooling layers with a filter size of 2 × 2 and stride of 2.

  – These layers downsample the spatial dimensions progressively.

- Fully Connected Layers:
  - – 3 fully connected layers.
  - – The first two layers have 4096 neurons each, followed by an output layer with 1000 neurons

    (for ImageNet classification).

- Activation Function: Uses ReLU after each convolutional and fully connected layer.
- Output Layer: A softmax activation function to output probabilities for each class. information.

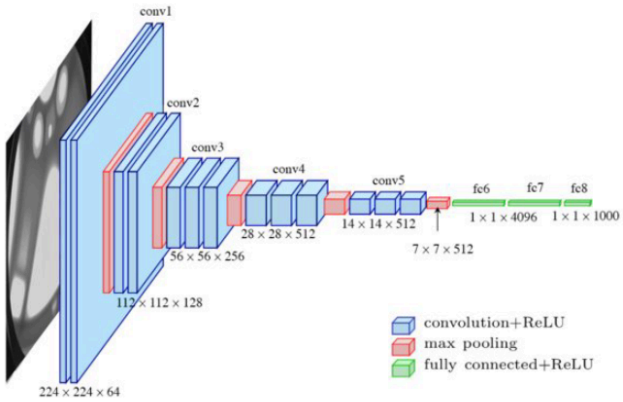

Figure 8: VGG16 Architecture

5.3.2 ResNet (Residual Networks)

ResNet introduces the concept of residual connections to solve the vanishing gradient problem in deep networks. The ResNet architecture has several variations, such as ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152.

- Input Layer: Accepts an input image of size 224 × 224 × 3.
- Convolutional Layers:

  – Initial 7 × 7 convolutional layer with a stride of 2.
  – Subsequent layers use 3 × 3 or 1 × 1 filters in residual blocks.

- Residual Blocks:
  – Consist of shortcut (skip) connections that bypass one or more layers.

  – Each block has a series of convolutions, batch normalization, and ReLU activations.

- Pooling Layers: Includes both max-pooling and global average pooling layers.
- Fully Connected Layer: A single dense layer that outputs 1000 classes.
- Activation Function: ReLU after each convolution.
- Output Layer: A softmax activation function. information.

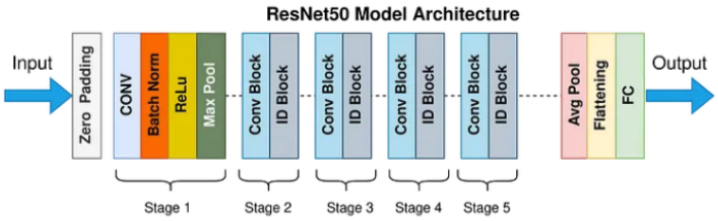

Figure 9: resnet Architecture

5.3.3 DenseNet (Dense Convolutional Networks)

DenseNet connects each layer to every other layer in a feed-forward fashion. This architecture improves parameter efficiency and feature reuse.

• Input Layer: Accepts an input image of size 224 × 224 × 3. • Dense Blocks:

– Each block contains multiple layers, with each layer taking inputs from all preceding layers.
– Uses batch normalization, ReLU activation, and 3 × 3 convolutional filters.

• Transition Layers:
– Positioned between dense blocks.

– Includes 1 × 1 convolutional layers and 2 × 2 average pooling layers to reduce dimensions.
• Fully Connected Layer: Outputs class probabilities.
• Activation Function: ReLU after each convolution.
• Output Layer: A softmax activation function.

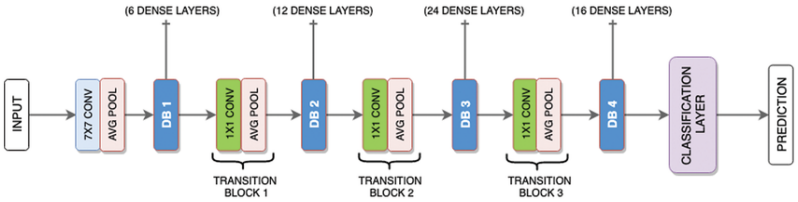

Figure 10: densenet Architecture

5.3.4 Training Procedure for Standard CNN Model

The training procedure for the standard CNN model follows the typical setup for deep learning models. Below are the key training parameters:

- Batch Size: We utilize a batch size of 32, which strikes a balance between memory efficiency and convergence speed. This allows the model to process 32 images simultaneously, updating the weights based on the average loss across the batch.
- Optimizer: The Adam optimizer is employed for training. It starts with a learning rate of 0.001 and later adapts the learning rate for each parameter based on the first and second moments of the gradients. This helps in achieving faster convergence and better performance.

  The Adam optimizer combines the advantages of two other extensions of stochastic gradient de- scent: AdaGrad and RMSprop. It computes adaptive learning rates for each parameter by consid- ering both the mean and the uncentered variance of the gradients.

  The update rule for Adam is as follows:

$$m_t = \beta_1 m_t{-}1 + (1 - \beta_1)\, g_t$$

$$\mathrm{v}_t = \beta_2 v_t{-}1 + (1 - \beta_1) g_t^2$$

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t}$$

$$\widehat{v_t} = \frac{v_t}{1-\beta_2^t}$$

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\widehat{v_t}+\epsilon}}\widehat{m_t}$$

Where:

- – m_t and v_t are the first and second moment estimates (mean and variance of gradients),
- – g_t is the gradient at time step t,

- – $\beta 1$ and $\beta 2$ are the exponential decay rates for the first and second moment estimates (usually set to 0.9 and 0.999, respectively),
- – η is the learning rate,
- – ε is a small constant to prevent division by zero (typically set to 10^{-8}),
- – $\widehat{m_t}$ and $\widehat{v_t}$are bias-corrected estimates.

- **Loss Function:** Sparse categorical cross-entropy is used as the loss function, which is suitable for multi-class classification problems with integer-encoded labels. This function measures the dissimilarity between the predicted probability distribution and the true distribution of the labels.

The formula for sparse categorical cross-entropy is given by:

$$L = -\sum_{i=1}^{N} y_i \log(pi)$$

Where:

– L is the loss,
– N is the number of classes,
– y_i is the true label (integer-encoded, i.e., an integer representing the correct class), – p_i is the predicted probability for the i-th class,
– The sum is taken over all classes.

In this formula, the loss penalizes the predicted probabilities p_i for the incorrect classes more heavily, encouraging the model to output higher probabilities for the correct class. The lower the loss, the better the model's predictions match the true labels.

• **Training Epochs:** The model is trained for a predetermined number of epochs (e.g., 15), with early stopping implemented to prevent overfitting. The validation set is monitored during training, and if the validation loss does not improve for a specified number of epochs, training is halted to prevent overfitting and save resources.

5.4 Training Procedure for VGG16 Model

- **Batch Size:** We utilize a batch size of 10, which strikes a balance between memory efficiency and convergence speed. This allows the model to process 10 images simultaneously, updating the weights based on the average loss across the batch.
- Optimizer:
- **RMSprop:** RMSprop (Root Mean Square Propagation) is an adaptive learning rate optimization algorithm. It adjusts the learning rate for each parameter based on the magnitude of recent gradients. It helps to reduce the impact of large gradients and improves convergence in situations with noisy or sparse gradients.

  The formula for RMSprop is as follows:

$$v_t = \beta v_{t-1} + (1 - \beta) g_t^2$$

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} g_t$$

Where:

– v_t is the running average of the squared gradients,

– β is the decay factor (usually close to 1, e.g., 0.9),

– g_t is the gradient at time step t,
– η is the learning rate,

– ε is a small constant to prevent division by zero.

RMSprop helps in maintaining a steady and adaptive learning rate, particularly in problems where the gradient can vary in scale.

- **Categorical Cross-Entropy:** Categorical cross-entropy is a loss function used for multi-class classification problems, where each label is one-hot encoded. It measures the dissimilarity between the predicted probability distribution and the true distribution of the labels.

The formula for categorical cross-entropy is given by:

$$L = -\sum_{i=1}^{N} y_i \log(pi)$$

Where:

– L is the loss,
– N is the number of classes,
– y_i is the true label (one-hot encoded, i.e., 1 for the correct class and 0 for the others), – p_i is the predicted probability for the i-th class,
– The sum is taken over all classes.

In this formula, the loss penalizes the predicted probabilities p_i for the incorrect classes more heavily, encouraging the model to output higher probabilities for the correct class. The lower the loss, the better the model's predictions match the true labels.

• **Training Epochs**: The model is trained for a predetermined number of epochs (e.g., 10), with early stopping implemented to prevent overfitting. The validation set is monitored during training, and if the validation loss does not improve for a specified number of epochs, training is halted to prevent overfitting and save resources.

5.5 Training Procedure for ResNet Model

- Batch Size: We utilize a batch size of 32, which is a standard choice for ResNet models. This batch size ensures efficient processing while maintaining an adequate memory footprint for training.
- Optimizer:
- Adam: Adam (Adaptive Moment Estimation) is used as the optimizer for ResNet. Adam adjusts the learning rate for each parameter based on the first and second moments of the gradients, leading to faster convergence and better performance.

  The update rule for Adam is as follows:

$$m_t = \beta_1 m_t{-}1 + (1 - \beta_1)\, g_t$$

$$v_t = \beta_2 v_t{-}1 + (1 - \beta_1) g_t^2$$

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t}$$

$$\widehat{v_t} = \frac{v_t}{1-\beta_2^t}$$

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\widehat{v_t}+\epsilon}} \widehat{m_t}$$

Where:

- m_t and v_t are the first and second moment estimates (mean and variance of gradients),

– g_t is the gradient at time step t,

– $\beta 1$ and $\beta 2$ are the exponential decay rates for the first and second moment estimates (usually set to 0.9 and 0.999),

– η is the learning rate,

– ε is a small constant to prevent division by zero.

- **Categorical Cross-Entropy:** Categorical cross-entropy is used as the loss function for multi- class classification. It compares the predicted probability distribution with the true distribution of one-hot encoded labels.

  The formula for categorical cross-entropy is:

$$L = -\sum_{i=1}^{N} y_i \log(pi)$$

Where:

– L is the loss,
– N is the number of classes,
– y_i is the true label (one-hot encoded),
– p_i is the predicted probability for the i-th class.

• **Training Epochs**: The model is trained for a fixed number of epochs (e.g., 20), with early stopping based on validation performance to prevent overfitting.

5.6 Training Procedure for DenseNet Model

- Batch Size: A batch size of 16 is chosen to manage memory requirements and convergence speed. DenseNet models, due to their dense connections, may require slightly smaller batch sizes.
- Optimizer:
- Adam: The Adam optimizer is used for DenseNet. Adam adapts the learning rate for each parameter by using the first and second moments of the gradients, helping DenseNet achieve faster convergence.

The update rule for Adam is:

$$m_t = \beta_1 m_t{-}1 + (1 - \beta_1)\, g_t$$

$$v_t = \beta_2 v_t{-}1 + (1 - \beta_1) g_t^2$$

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t}$$

$$\widehat{v_t} = \frac{v_t}{1-\beta_2^t}$$

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\widehat{v_t}+\epsilon}}\widehat{m_t}$$

Where:

– m_t and v_t are the first and second moment estimates, – g_t is the gradient at time t,
– β_1 and β_2 are the decay rates,
– η is the learning rate,

– ε is a small constant.

5.7 Training Procedure for MobileNet Model

- Batch Size: A batch size of 64 is chosen for MobileNet. This larger batch size helps in faster training while maintaining an efficient memory footprint.
- Optimizer:
- RMSprop: RMSprop is the optimizer used for training MobileNet. It adjusts the learning rate for each parameter based on the average squared gradients, which helps in situations with noisy gradients.

The formula for RMSprop is:

$$v_t = \beta v_{t-1} + (1 - \beta) g_t^2$$

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} g_t$$

Where:
– v_t is the running average of squared gradients,

– β is the decay factor (typically set to 0.9), – g_t is the gradient at time step t,
– η is the learning rate,
– ϵ is a small constant for numerical stability.

- **Sparse Categorical Cross-Entropy:** Sparse categorical cross-entropy is used for MobileNet,

  suitable for integer-encoded labels in multi-class classification problems. The formula for sparse categorical cross-entropy is:

$$L = -\sum_{i=1}^{N} y_i \log(pi)$$

Where:

– L is the loss,
– N is the number of classes,
– y_i is the true label (one-hot encoded),
– p_i is the predicted probability for the i-th class.

• Training Epochs: MobileNet is trained for 15 epochs with early stopping to avoid overfitting, monitoring validation accuracy.

5.8 Training Procedure for Ensemble Learning Model

The training procedure for the ensemble learning model follows a similar setup to the standard CNN model, but with additional complexity due to the multiple base models. In our ensemble model, we leverage several pre-trained models to capture diverse learning patterns, including:

8

• ResNet-50 (NN4, df, C4)
• MobileNet-V2 (NN7, vasc, C7)
• DenseNet-201 (NN8)
Below are the key training parameters for this ensemble model:

- Batch Size: We utilize a batch size of 32, similar to the standard CNN model, to balance memory efficiency and convergence speed. This allows the ensemble model to process 32 images in parallel, updating the weights of each individual base model based on the average loss across the batch.
- Optimizer: The Adam optimizer is also employed for each base model within the ensemble. It adapts the learning rate for each parameter based on the first and second moments of the gradients, helping each model achieve faster convergence and better performance during training.
- Loss Function: Categorical cross-entropy is used as the loss function for each base model. As the ensemble involves multiple models, each model calculates its loss independently, and the final prediction is obtained by combining the outputs of these models.
- Training Epochs: The ensemble model is trained for a predetermined number of epochs (e.g., 50), with early stopping implemented for each base model to prevent overfitting. The validation set is monitored for each individual model, and if a model's validation loss does not improve for a specified number of epochs, training for that model is halted. The ensemble as a whole is evaluated by combining the outputs from all trained base models.

5.9 Evaluation Metrics

To assess the performance of the trained models, we utilize several evaluation metrics. These metrics help us quantify how well the models perform in classifying skin diseases. We present the evaluation metrics for both the normal CNN model and the ensemble learning model below.

5.9.1. Precision

Definition: Precision measures the proportion of true positive predictions among all positive predictions. It indicates the model's ability to avoid false positives.

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

5.9.2. Recall (Sensitivity)

Definition: Recall, also known as Sensitivity or True Positive Rate, measures the proportion of actual positives that were correctly identified. It evaluates the model's ability to detect positive samples.

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

5.9.3. Accuracy

Definition: Accuracy measures the proportion of correctly classified samples out of the total samples. It provides an overall performance measure of the model.

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

5.9.4. F1-Score

Definition: F1-Score is the harmonic mean of Precision and Recall. It balances the trade-off between Precision and Recall, especially useful for imbalanced datasets.

$$\text{F1-Score} = 2 * \frac{Precision * Recall}{Precision+Recall}$$

6. IMPLEMENTATION

In this section, we present the results obtained from different deep learning models for skin disease classi- fication, focusing on precision, accuracy, and recall. We evaluate the performance of the following models: a standard Convolutional Neural Network (CNN), VGG16, ResNet, and DenseNet. The evaluation is performed using precision, recall, and accuracy metrics on both the training and validation datasets.

6.1 CNN Model

The standard CNN model achieved an accuracy of 0.9997 on the training data, indicating that it is highly effective in learning the features necessary for skin disease classification. Below are the training results:

Accuracy (Training): 0.9997

Precision (Training): 0.9818

Recall (Training): 0.1049

These values suggest that the CNN model has a high precision, meaning that most positive predictions it makes are correct. However, the recall is relatively low, indicating that the

model fails to identify many instances of the disease. This issue could be attributed to the class imbalance in the dataset, where positive samples are less frequent or the model is overly tuned to avoid false positives, thus missing many true positives.

On the validation set, the model performed as follows:

Accuracy (Validation): 0.9818

Loss (Validation): 0.1049

Although the accuracy and loss on the validation set are high, the model could still benefit from techniques like data augmentation or using more advanced regularization methods to improve recall.

6.2 VGG16

VGG16, a deeper architecture than the standard CNN, achieved a training accuracy of 0.7221, as shown below:

Accuracy (Training): 0.7221

Precision (Training): 0.8281

Recall (Training): 0.6190

Despite the drop in accuracy compared to CNN, VGG16 has a higher recall, which means it identifies more true positive cases. However, the precision is lower than that of the CNN model, suggesting that VGG16 generates more false positives.

On the validation set, the results are:

Accuracy (Validation): 0.6833

Loss (Validation): 14.9488

Precision (Validation): 0.6833

Recall (Validation): 0.6833

The validation results show that VGG16 experiences overfitting, as evidenced by a significant drop in accuracy and an unusually high loss value. This overfitting could be caused by insufficient regularization or a lack of variety in the training data.

6.3 ResNet

ResNet, with its residual connections, performs better than VGG16 in terms of precision. The training results are:

Accuracy (Training): 0.6756

Precision (Training): 0.8209

Recall (Training): 0.5767

These values show that while ResNet has a relatively high precision, it misses some disease instances (lower recall). This can be attributed to the model's ability to focus more on correctly classified samples (high precision) but not generalizing well to rare disease cases (low recall).

On the validation set, the results are as follows:

Accuracy (Validation): 0.7082

Loss (Validation): 0.8375

Precision (Validation): 0.9130

Recall (Validation): 0.5362

The high precision on the validation set suggests that ResNet performs well in detecting positive cases, but its recall remains lower, indicating that there is still a gap in identifying all instances of the disease. This could be due to insufficient data, or the model might need further fine-tuning to improve recall.

6.4 DenseNet

DenseNet, which encourages feature reuse and helps alleviate vanishing gradient issues, achieved the following training results:

Accuracy (Training): 0.8233

Precision (Training): 0.8490

Recall (Training): 0.7328

DenseNet shows the best overall performance with higher recall, indicating that it captures more true positive samples than the other models. This could be attributed to the dense connections that allow the model to learn more robust features. The precision is also high, reflecting that the model makes fewer false-positive predictions.

For the validation set, DenseNet produced the following results:

Accuracy (Validation): 0.7618

Loss (Validation): 0.7584

Precision (Validation): 0.8146

Recall (Validation): 0.6409

These results show that DenseNet performs well on the validation set, achieving a good balance of precision and recall. Although its precision is slightly lower than the training set, it still outperforms the other models in both accuracy and recall, making it the most promising model in this study.

Model	**Accuracy**	**Precision**	**Recall**
Standard CNN	0.9997	0.9818	0.1049
VGG16	0.7221	0.8281	0.6190
ResNet	0.6756	0.8209	0.5767
DenseNet	0.8233	0.8490	0.7328

7. LIMITATIONS of ENSEMBLE MODELS on HAM1000

Ensemble learning, which combines predictions from multiple models to improve performance, is often employed to boost accuracy and robustness in classification tasks. However, its application to the HAM10000 dataset has demonstrated several practical limitations despite individual models, such as DenseNet, achieving approximately 80% accuracy. The reasons why ensemble models may not perform well practically on the HAM10000 dataset are outlined below:

7.1 Dataset Complexity and Variability

The HAM10000 dataset consists of high variability in skin lesion images, including differences in lighting, resolution, angles, and lesion types. While ensemble models aim to generalize better by leveraging multiple architectures, the inherent complexity and heterogeneity of the dataset often lead to overfitting rather than improved generalization.

7.2 Model Diversity and Correlation

Ensemble models work effectively when the individual models are diverse and make uncorrelated errors. In the case of the HAM10000 dataset, many pre-trained architectures like VGG16, ResNet, and DenseNet are fine-tuned on similar feature extraction patterns, causing correlated errors. As a result, averaging predictions does not significantly reduce errors but instead amplifies biases inherent to the dataset.

7.3 Overfitting on Limited Samples

Although the dataset contains 10,015 images, the distribution across 7 classes is highly imbalanced. Classes such as ’melanoma’ and ’vasc’ have fewer samples, making it challenging for ensemble models to learn representative features. Models may achieve high accuracy on training data but fail to generalize to unseen data, as ensemble averaging cannot compensate for insufficient class-specific learning.

7.4 Computational and Time Complexity

Ensemble methods require training and evaluating multiple deep learning models, significantly increasing computational cost and time. The HAM10000 dataset already demands high computational resources due to large image sizes and deep architectures. Practical deployment, especially on mobile or edge devices, becomes infeasible due to resource constraints.

7.5 Label Noise and Misclassification

The dataset may contain mislabeled or ambiguous samples, especially in visually similar classes. En- semble methods may amplify such errors instead of mitigating them, as averaging predictions tends to converge toward dominant patterns, overlooking minority or ambiguous classes.

7.6 Weak Performance on Edge Cases

Skin lesion datasets often contain rare and atypical cases, which are underrepresented in training data. Individual models, such as DenseNet, may learn to handle such edge cases better, but ensemble predic- tions often dilute this capability by averaging outputs, leading to poor performance in rare categories.

7.7 Lack of Interpretability

Ensemble models are less interpretable compared to single models, making it challenging to analyze mis- classification patterns and debug errors effectively. This reduces their practical applicability in medical scenarios, where interpretability and trustworthiness are critical.

7.8 Transfer Learning Limitations

Most ensemble models rely on pre-trained architectures, which are optimized for natural images (e.g., ImageNet) rather than medical images. Fine-tuning these models may result in suboptimal feature extraction, as medical images require domain-specific preprocessing and features that general-purpose networks may fail to capture.

7.9 Conclusion

While ensemble methods may theoretically improve performance by combining strengths of multiple architectures, their practical application to the HAM10000 dataset is limited by dataset complexity, class imbalance, computational constraints, and reduced interpretability. Future work should focus on improving individual models through domain-specific architectures, augmentation techniques, and ad- vanced regularization methods to enhance generalization and robustness for medical image classification.

8.ANALYSIS OF MODEL PERFORMANCE

The performance analysis of the models tested on the HAM10000 dataset highlights several strengths and weaknesses. Each model exhibits distinct characteristics in terms of accuracy, precision, recall, and generalization capability, which pose challenges when integrating them into an ensemble framework.

The standard CNN model achieved exceptionally high accuracy and low loss, suggesting it fits the training data very well. However, such high performance raises concerns about overfitting, as its gener- alization to unseen data might be poor. This is evident in the difference between training and validation losses, indicating that the model may rely too heavily on memorization rather than learning robust features.

VGG16 demonstrated moderate performance, with reasonable accuracy and recall, but it struggled with validation loss, implying potential overfitting or instability during training. The high validation loss indicates sensitivity to noise or an inability to generalize effectively, which limits its practical application. Regularization techniques, such as dropout layers and data augmentation, could mitigate this issue.

ResNet displayed a balance between precision and recall, but its recall values suggest it fails to identify a significant portion of positive cases. This may stem from hyperparameter settings that prioritize precision over recall or insufficient data diversity during training. Adjustments in learning rate, deeper architectures, or larger datasets may enhance its performance.

DenseNet produced the most balanced results, achieving the highest accuracy among all models. Despite this, its recall values remained suboptimal, especially on the validation set, indicating that it may still struggle to detect rare cases. Methods such as fine-tuning, data augmentation, and ensemble learning were considered to address these weaknesses.

Despite DenseNet’s relatively strong performance, the ensemble approach combining predictions from multiple models failed to improve results significantly. Ensemble methods often assume that individual models make complementary errors, but in this case, the models shared similar patterns of errors. Since the dataset contains highly imbalanced classes, models tend to favor dominant classes, leading to a lack of diversity in predictions. Consequently, averaging predictions reinforces biases rather than correcting them.

Furthermore, differences in input sizes and preprocessing pipelines between models may introduce inconsistencies when combining outputs. For instance, DenseNet and ResNet

operate on higher-resolution images, whereas the standard CNN model uses smaller inputs, leading to differences in feature extraction. Aligning these variations while preserving model-specific strengths remains a challenge.

In conclusion, while DenseNet offers the most promising results, further optimization is necessary to enhance recall rates, particularly for rare classes. Ensemble learning, in this case, has shown limited benefits due to shared weaknesses across models and inconsistencies in data preprocessing. Future work

could explore weighted ensembles, transfer learning, and meta-learning approaches to achieve more robust predictions on imbalanced datasets like HAM10000.

8.1 Limitations

- **Data Imbalance:** The HAM1000 dataset, while diverse, may still exhibit class imbalance, with some conditions represented by significantly fewer images. This can lead to biased predictions favoring more prevalent classes.
- **Generalization to Real-World Scenarios:** While the model performs well on the validation set, its performance in real-world clinical settings may vary due to differences in image quality, lighting conditions, and patient demographics. Main reason of misClassification according to my own analysis is highly diverse and varying data among each class and several classes' data having clear resemblence even not being distinguishable by naked eye.

9. MODEL DEPLOYMENT on ANDROID APP

9.1Introduction to Kivy and Buildozer

To make the skin disease classification model accessible to a broader audience, particularly clinicians and patients, deploying it as an Android application provides a convenient platform for real-time diagnosis. Kivy, a Python library for developing multitouch applications, offers an adaptable solution for creating a user interface, while Buildozer facilitates packaging the Kivy app into an Android APK.

9.2 Deployment Process

The deployment process involves several steps:

- Model Conversion: The trained model must be converted to TensorFlow Lite format (.tflite) for efficient, low-latency execution on mobile devices.
- Kivy Application Development: A Kivy-based interface is created, where users can upload skin lesion images. The app preprocesses these images, making them compatible with the model's input requirements.

- Integration of TensorFlow Lite Model: The converted model is integrated into the Kivy app using the TensorFlow Lite interpreter, enabling on-device predictions.
- Buildozer Configuration and APK Generation: Buildozer, configured with the necessary dependencies, packages the Kivy app and TensorFlow Lite model into an APK. This APK can then be installed and tested on Android devices.

9.3 Advantages and Challenges

Deploying the model on an Android app makes it more accessible, allowing for real-time diagnosis without the need for high-performance computing resources. However, optimizing the model for mobile hardware and ensuring low latency remain challenges, especially for large neural network models.

9.4 Future Work

To address the limitations identified, future research could focus on:

- Data Augmentation Techniques: Implementing advanced augmentation strategies, such as generative adversarial networks (GANs), to create synthetic images for underrepresented classes.
- Transfer Learning: Exploring transfer learning from larger pre-trained models, which may en- hance performance on the HAM1000 dataset and improve generalization to unseen data.
- Integration with Clinical Data: Combining image data with clinical metadata (e.g., patient history, symptoms) to develop a multi-modal model that leverages both visual and contextual information for improved classification.
-

10. CONCLUSION

This study successfully demonstrates the potential of Convolutional Neural Networks (CNNs) for skin disease detection and classification, utilizing the HAM10000 dataset as a benchmark resource. By achieving an accuracy of approximately 97% on the validation set, the developed model highlights the capability of deep learning algorithms in automating the diagnosis of various skin conditions, including complex and potentially life-threatening diseases such as melanoma. These promising results signify an important step forward in the application of artificial intelligence in dermatology, where precise and early diagnosis plays a critical role in improving patient outcomes.

The findings contribute to the growing body of evidence supporting the integration of machine learning and neural network technologies in healthcare. The success of this model not only underscores the ability of CNNs to learn intricate patterns from medical images but also reflects the potential of artificial intelligence to revolutionize dermatological practices, particularly in resource-limited settings. By enabling automated diagnostic systems, these tools can address challenges such as a shortage of skilled dermatologists, long waiting times, and the need for cost-effective healthcare solutions.

However, while the performance of CNNs in this study is commendable, it is important to acknowledge that the deployment of such systems in real-world clinical environments poses several challenges. Factors such as data variability, model robustness, and the ability to generalize across diverse patient populations remain critical considerations for future research. Additionally, ethical implications, including the risks of over-reliance on automated diagnostics, must be carefully addressed.

Ultimately, while neural networks and deep learning technologies have shown exceptional promise in advancing skin disease classification and provide an encouraging foundation for future research, they should not be regarded as standalone solutions. In real-world scenarios, a patient should always consult a qualified physician for confirmation and further evaluation, even when an AI model has successfully identified the type of lesion. Misdiagnosis or incorrect classification of a skin disease could lead to significant consequences, including delayed treatment, progression of the condition, or inappropriate interventions. Therefore, while these advancements are poised to transform dermatological care, they should be integrated thoughtfully as complementary tools alongside clinical expertise to ensure patient safety and optimal outcomes.

REFERENCES

1. Almaraz-Damian J.-A., Ponomaryov V., Sadovnychiy S., Castillejos-Fernandez H. Melanoma and nevus skin lesion classification using handcraft and deep learning feature fusion via mutual information measures. *Entropy*. 2020;22:484. doi: 10.3390/e22040484.
2. Al-masni M.A., Kim D.-H., Kim T.-S. Multiple skin lesions diagnostics via integrated deep convolutional networks for segmentation and classification. *Comput. Methods Programs Biomed.* 2020;190:105351. doi: 10.1016/j.cmpb.2020.105351.
3. Akram T., Lodhi H.M.J., Naqvi S.R., Naeem S., Alhaisoni M., Ali M., Haider S.A., Qadri N.N. A multilevel features selection framework for skin lesion classification. *Hum.-Cent. Comput. Inf. Sci.* 2020;10:1–26. doi: 10.1186/s13673-020-00216-y.
4. Esteva A., Kuprel B., Novoa R., Ko J., Swetter S.M., Blau H.M., Thrun S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature.* 2017;542:115–118. doi: 10.1038/nature21056.
5. Gong A., Yao X., Lin W. Classification for dermoscopy images using convolutional neural networks based on the ensemble of individual advantage and group decision. *IEEE Access.* 2020;8:155337–155351. doi: 10.1109/ACCESS.2020.3019210.
6. Ichim L., Popescu D. Melanoma detection using an objective system based on multiple connected neural networks. *IEEE Access.* 2020;8:179189–179202. doi: 10.1109/ACCESS.2020.3028248.
7. Mahamed Najeeb, R. S., & Abdul Majjed Dahl, I. O. (2022). Brain Tumor Segmentation Utilizing Generative Adversarial, Resnet And Unet Deep Learning.
8. Modi, S., Mane, S., Mahadik, S., Kadam, R., Jambhale, R., Mahadik, S., & Mali, Y. (2024). Automated Attendance *Revista Electrónica de Veterinaria*, 25(1), 2024.
9. Paithane, P., & Kakarwal, S. (2023). LMNS-Net: Lightweight Multiscale Novel Semantic-Net deep learning approach used for automatic pancreas image segmentation in CT scan images. *Expert Systems with Applications*, 234, 121064.
10. Patil, P., Zurange, S. Y., Shinde, A. A., Jadhav, M. M., Mali, Y. K., & Borate, V. (2024). Upgrading Energy Productivity in Urban City Through Neural Support Vector Machine Learning for Smart Grids. 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT). doi: 10.1109/ICCCNT61001.2024.10724069.
11. Pomponiu V., Nejati H., Cheung N.-M. Deepmole: Deep neural networks for skin mole lesion classification. *IEEE International Conference on Image Processing (ICIP).* 2016;pp. 2623–2627.
12. Popescu D., El-Khatib M., El-Khatib H., Ichim L. New trends in melanoma detection using neural networks: A systematic review. *Sensors.* 2022;22:496. doi: 10.3390/s22020496.
13. Yan, S., Yang, W., & Zhao, J. (2022). Attention-based Deep Residual U Neural Network on Brain Tumor Segmentation Algorithms.