

Abstract

Skin cancer is one of the most dangerous diseases that, if not diagnosed early, poses a high risk of spreading to other areas of the body, thus increasing mortality rates. Traditionally, the classification of skin cancer lesions done manually by dermatologists, is time consuming and subjective. But with the advancements in Computer Vision technology, deep learning models are playing a crucial role in early detection and treatment of skin cancer. In the past studies in this field, traditional CNN approaches were used without any data augmentation, dimensionality reduction and advanced modelling techniques, which resulted in accuracy levels around 77%. This study aims to address these challenges associated with high-dimensional data and image inconsistencies through a comprehensive pre-processing strategy. Techniques such as MinMaxScaler for data standardization, KNN imputation for missing values, the Dull Razor method for image cleaning, L2 regularization, Principal Component Analysis (PCA) for effective dimensionality reduction and data augmentation were employed. This study employs advanced deep learning models including Convolutional Neural Network (CNN), CNN with One-vs-All (CNN OVA), ResNet50, and DenseNet121, each subjected to meticulous hyperparameter tuning. The ResNet50 achieved the highest accuracy of 99.61%, significantly outperforming other models DenseNet121 at 95%, CNN OVA at 97.72%, and CNN at 94%. This demonstrated higher accuracy underscores the importance of pre-processing techniques, advanced transfer learning methods, hyperparameter tuning and data augmentation in skin cancer lesions classification, paving new paths for advancements in clinical diagnostics.

Introduction

Background

According to Cancer.Net, skin cancer is caused by abnormal growth of skin cells. These abnormal cells can form tumors, which can be cancerous, that means they can spread to other parts of the body, or benign that can grow but not spread. Skin cancer can be caused by many factors. The most significant risk factor is exposure to ultraviolet (UV) radiation from the sun. People with fair skin, weakened immune systems, or a history of sunburn are especially susceptible. Others include artificial UV exposure from tanning beds, certain medical conditions, medications, and genetic predispositions.

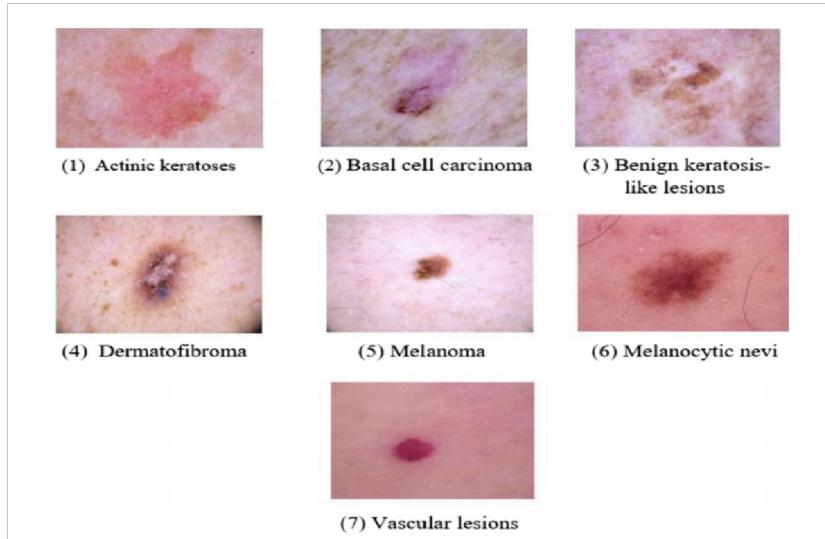
Tschandl et al. (2018) described the classification of skin cancer including Actinic keratoses, Basal cell carcinoma, Benign Keratosis, Dermatofibroma, Melanocytic nevi, Melanoma and Vascular skin lesions. Actinic keratosis is a type of skin cancer caused by over exposure to the sun. It usually appears as a rough, scaly patch and is not dangerous because it doesn't spread deeper into the skin. These types of lesions can be treated easily by a doctor without the need for surgery. They are found mostly on the face, where sun exposure is strongest. Benign keratosis is a broad category for various non-cancerous skin growths. This includes seborrheic keratosis (often called "senile warts"), solar lentigos (age spots), and lichen planus-like keratoses (inflamed seborrheic keratoses or age spots). These growths, although appearing different, are grouped together because they behave similarly on a biological level and are often diagnosed using the same tissue examination. They typically have a rough, scaly, and wart-like texture, with colors ranging from light brown to black. Vascular skin lesions are harmless bumps caused by blood vessels. These lesions range from cherry angiomas and angiokeratomas to pyogenic granulomas. When examined under a special magnifying tool used by dermatologists, these lesions typically appear red or purple and have a solid, well-defined structure. Melanocytic nevi are a type of mole and come in many variations. An

important difference between a nevus and melanoma (a cancerous mole) is that nevi are usually symmetrical in colour and structure throughout the mole.

Zaballos et al. (2008) explained that dermatofibromas are non-cancerous growths that appear mostly on the young or middle-aged adults, particularly women. These firm bumps, usually a few millimeters to 2 centimeters wide, can appear anywhere but are especially common on the lower legs. They come in various colors like light brown, dark brown, purple-red, or even yellow. These growths arise from an overgrowth of fibrous tissue and sometimes white blood cells in the deeper layers of the skin, and although harmless, they can be itchy or irritating. Basal cell carcinoma (BCC) affects both fair-skinned and darker-skinned people. It typically appears as a flesh-coloured bump, pearly nodule, or pinkish patch after years of sun exposure or indoor tanning. While most common on the head, neck, and arms, BCC can develop anywhere on the body. Early detection and treatment are crucial because BCC can grow deep and cause damage if left untreated. Melanoma, considered the most serious form of skin cancer due to its ability to spread, can develop from an existing mole or appear as a new, unusual dark spot on the skin. Early detection and treatment are critical for its treatment. To aid in early identification, doctors usually follow the ABCDE warning signs of melanoma: Asymmetry (one half unlike the other), Border (irregular, ragged), Color (uneven, with multiple shades), Diameter (larger than 6 millimeters), and Evolving (changing in size, shape, or color). The seven classes of skin cancer are shown in the Figure 1.

Figure 1

Sample images of seven classes of skin cancer lesions



Note: This picture is taken from Shetty et al. (2022)

For this project, utilized the HAM10000 dataset (Tschandl et al., 2018), which is collected from Harvard Dataverse and is stored in storage buckets in Google Cloud Platform for ease of accessibility and security. On this data, several data preprocessing and data augmentation techniques were employed to enrich the dataset for better modelling. These include data cleaning techniques like imputing any missing values, encoding categorical features, removing hair from images using dull razor method (Gururaj et al., 2023) and augmenting data to balance the underrepresented classes in the dataset.

Traditional methods of detecting the class of skin cancer involve visually examining the skin and conducting a skin biopsy to test suspicious areas for cancer, aiding in determining the type of skin cancer present. Additional tests may be necessary to evaluate the extent or stage of the cancer, particularly for more advanced cancers like squamous cell carcinoma or melanoma, which may require imaging tests or lymph node examination (Skin Cancer - Diagnosis and Treatment - Mayo Clinic, 2022). Early detection of skin cancer early is crucial as it can significantly impact the

chances of successful treatment. (Common Skin Cancer Can Signal Increased Risk of Other Cancers, 2018) Basal cell carcinoma and squamous cell carcinoma, the most common forms of skin cancer, are often curable when detected and treated early, with a lower likelihood of spreading extensively to other parts of the body. In contrast, melanoma poses a higher risk of spreading to nearby tissues and beyond, making it more challenging to cure if not identified and treated promptly. Failure to detect skin cancer early can lead to more advanced stages of the disease, increasing the complexity of treatment and potentially reducing the chances of successful outcomes. Therefore, it becomes necessary to detect the class of skin cancer accurately and early, to reduce the deaths caused by the disease.

Nachbar et al. (1994) suggests that although traditional machine learning methods can achieve good results in specific skin cancer classification tasks, their reliance on manual feature extraction limits their effectiveness in handling the complexities of real-world clinical diagnoses. These methods typically involve extracting features from skin lesion images, followed by classification. According to Guo et al. (2016), deep learning offers a significant leap forward. It bypasses the need for manual feature extraction and domain expertise altogether. Deep learning algorithms can analyse vast datasets much faster and more accurately than traditional methods, automatically extracting the most relevant characteristics crucial for skin cancer diagnosis. This not only streamlines the process but also empowers clinicians with more robust diagnostic tools. Overall, the application of deep learning in skin cancer detection offers a promising avenue for enhancing diagnostic capabilities and improving patient outcomes.

Convolutional Neural Networks

Fuadah et al. (2020) explained that Convolutional Neural Networks (CNNs) are specialized for image analysis, building upon traditional neural networks to handle 2D data. They are "deep" networks with many layers enabling them to learn complex patterns. Their unique architecture

includes several key components: the Convolution Layer extracts features using filters, the ReLU activation layer improves efficiency, and the Pooling Layer streamlines data to prevent overfitting. Finally, the Fully Connected Layer classifies the image. CNNs also use sophisticated optimizers like Adam or Nadam that refine the network's learning process, making it faster and more efficient. The CNN algorithm could help doctors make faster, more accurate diagnoses, addressing issues of timely treatment and improving overall healthcare outcomes. This model could then surpass specialists in identifying different skin cancers from images. This capability would be especially impactful in areas with limited access to dermatologists, where early detection is crucial to successful skin cancer treatment.

Convolutional Neural Networks– One-Vs-All (CNN-OVA)

The OVA-CNN approach divides the task of multiclass classification into multiple binary classification tasks, utilizing an ensemble of binary classifying CNNs (BCNNs) and a maximum-value selector. Comprising ten BCNNs, each trained to specialize in recognizing a single target class, the ensemble module generates varying probabilities for the same target. During testing, each BCNN predicts the probability of the input image belonging to its designated target class. The final classification decision is made by the max-value selector, which selects the class corresponding to the BCNN with the highest predicted probability. The structure of a BCNN closely resembles a conventional CNN, with convolution layers, pooling operations, batch normalization, ReLU activation, fully connected layers, and a SoftMax layer. Each BCNN reduces the number of neurons in its layers compared to a standard CNN to mitigate overfitting and accommodate training with fewer samples, thereby enhancing its generalization ability (Babu & Narayanan, 2022).

ResNet50

Residual Networks (ResNets) offer an innovative solution to a common issue in deep neural networks: degradation. As networks become deeper, their performance can surprisingly worsen

rather than improve. ResNets cleverly address this through "shortcut connections" that jump over layers. This allows crucial information from earlier layers to flow directly to later ones, combating the degradation problem and facilitating the training of deeper, more powerful networks.

ResNet50 is a 50-layer deep convolutional neural network trained on the massive ImageNet dataset, which contains over 14 million images across thousands of categories. Its architecture relies on convolutional layers with varying filter sizes (1x1 and 3x3) and is organized into five convolutional stages. The first stage (Conv1) has a single convolutional block, while the remaining stages feature multiple blocks with a repeated structure of three layers each. Feature map size is down sampled using average pooling. Finally, a fully connected convolutional layer handles the classification task. This specific 50-layer design leads to the name ResNet50 (Polat & Gungen, 2021).

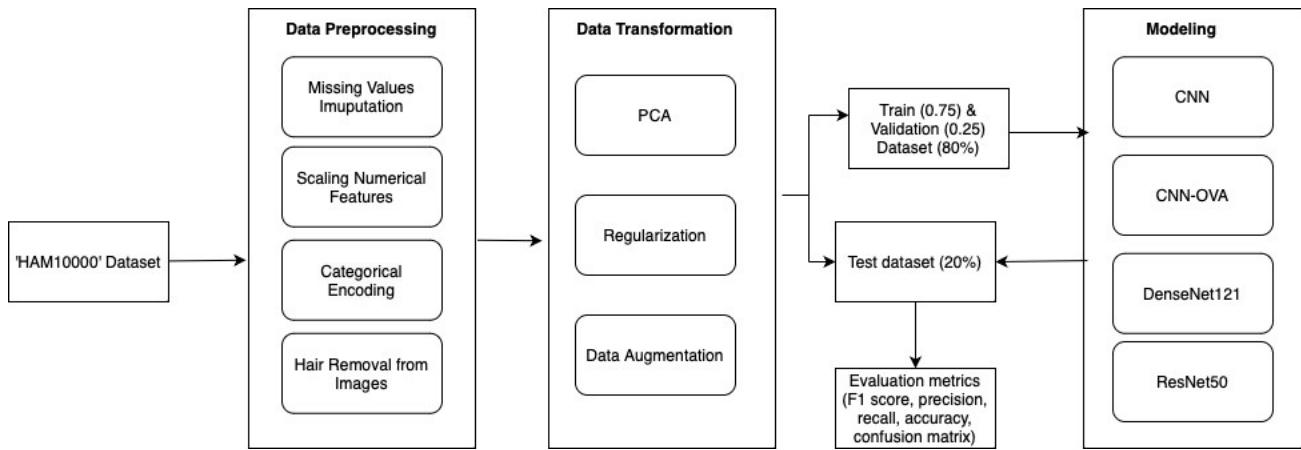
DenseNet121

Huang and colleagues (2016) described DenseNet, a novel deep learning structure that builds on ResNet's concepts but adopts a distinct layer connection approach. Rather than ResNet's approach of adding outputs, DenseNet combines the output of each layer with subsequent layers. This design ensures that each DenseNet layer receives all previous layers' feature maps, enhancing the flow of information. Consequently, DenseNet can deliver robust performance with a reduced number of layers, exemplified by DenseNet121's 121-layer structure. Comprising blocks of convolutional layers, DenseNet utilizes all preceding layers' feature maps within a block as inputs for subsequent layers. This architecture has shown marked improvements over other leading methods, achieving high performance with less computational demand.

Figure 2 provides an overview of the entire project workflow.

Figure 2

Project Workflow



The developed model will be further refined to enhance its accuracy through image preprocessing techniques such as rotation, resizing, flipping, and grayscale conversion. Extensive hyperparameter tuning, employing methods like grid search or random search, will optimize model performance, considering factors such as learning rate and regularization strength. Evaluation metrics such as F1 Score, recall, precision, accuracy, and confusion matrix will assess model performance. Once the model is fully developed, they will be integrated into clinical systems or specialized applications for accurate skin lesion diagnosis. Leveraging advanced preprocessing techniques and robust model architectures, the system facilitates early cancer detection, with real-time data evaluation on a web platform to gather feedback from medical professionals. This can help them detect skin cancer at early stages and provide treatment before it spreads to other parts of the body, reducing the mortality rate. Overall, the utilization of deep learning in skin cancer detection presents a hopeful opportunity to enhance diagnostic capabilities and patient outcomes.

Project Requirements

Functional Requirements

The suggested Skin Cancer Classification requires implementation of deep neural networks like CNN (Convolutional Neural Network), CNN-OVA (Convolutional Neural Network – One Versus

All), ResNet50 and DenseNet121. For dimensionality reduction of images utilizing Principal Component Analysis (PCA) and GANs are employed to diversify and increase the dataset which provides more robust data for model training.

Principal Component Analysis (PCA) (Elgamal, M., 2013) for reduction of high dimensional data to lower-dimensional space. It is the most used statistical technique for compression of images. Dataset HAM10000 is limited, to increase the dataset and generate more realistic images GANs (Ahmad, B. et al., 2021) are implemented after the reduction of the dimensionality of the images. GANs are composed of two components: generator and discriminator. Generator generates the images which resemble the images in the training dataset and discriminator checks the images which are generated by the generator whether they are fake or not and gives feedback to the generator. This process continues until the generator is capable of generating more realistic images. Implementing this process helps to improve the accuracy of implementing models by exposing the models to a broad spectrum of variations in the data.

The training includes models like CNN (Convolutional Neural Network) (Esteva et al., 2017), CNN-OVA (Convolutional Neural Network – One Versus All) (Polat, Kemal & Koc, Kaan., 2020), ResNet50 (Ö. Polat & Güngen, 2021) and DenseNet121 (Al-Saedi & Savaş, 2022), which are optimized through algorithms like Adam or Stochastic Gradient Descent (SGD). The training process is continued until optimal performance is achieved.

Integrated nonlinear activation functions like ReLU (Naqvi et al., 2023), LeakyReLU (Aldhyani et al., 2022), and ELU (Gayatri & Aarthy, 2023) with CNN and CNN-OVA to improve learning of complex patterns. Post-training, the model's performance is evaluated by using accuracy, F1-score and confusion matrix metrics. To improve the performance, fine tune the models by changing the hyperparameters such as optimizers, drop out ratio, batch size, kernel size, padding. Again, evaluate the model's performance and compare it with the previous performance

before hyperparameter tuning. Building a simple user interface using streamlit as it takes input as an image and successfully classifies the skin lesion.

AI Requirements

Skin cancer classification models are developed, trained and deployed using TensorFlow, Keras, and sklearn libraries. Dimensionality reduction is performed using Principal Component Analysis (PCA) (Elgamal, 2013) from the sklearn library. In CNN (Esteva et al., 2017), the architecture was implemented using Keras, focusing on spatial feature extraction from skin lesion images. Model will undergo optimization and hyperparameter tuning to maximize classification accuracy.

For the CNN-OVA model (Polat & Koc, 2020) the architecture is implemented using TensorFlow. Each cancer type is treated as a separate binary classification problem, allowing the model to focus on identifying one specific cancer type against all others in each iteration. This method enhances the model's ability to recognize distinct features of each cancer type, leading to more accurate classification results. The CNN-OVA strategy effectively handles the multi-class nature of skin cancer types, ensuring precise detection and classification within the dataset.

ResNet50 (Polat & Gnguen, 2021) model is implemented by using TensorFlow, along with its high-level API Keras. Allows the training of very deep networks by using skip connections or shortcuts to jump over some layers. For DenseNet121 (Al-Saeid & Savas, 2022), the architecture is designed using TensorFlow. The architecture also facilitates feature reuse throughout the network and it also helps in reducing the number of parameters and computational efficiency while enhancing feature propagation and extraction.

All the models are evaluated using accuracy, F1-score and confusion matrix, utilizing the functionalities provided by the sklearn library. The implementation, training, evaluation of models and implementation of user interface are done in Google Colab Pro, making use of its integrated CPUs, GPUs and substantial RAM to meet the project's computational requirement.

Data Requirements

In this project, the data is derived from the HAM10000 dataset (Tschandl, 2018). It consists of 10015 dermatoscopic images and are distributed over seven classes. The images are of different resolutions and there are class imbalances. Each image has an id and associated metadata provided in the csv file. Data augmentation is helped to address class imbalances and improve the model's performance. The dataset is publicly available in Harvard Dataverse, and it can be downloaded from the original repository. This dataset can be served as training data for the academic machine learning projects as mentioned in the Harvard Dataverse website. The project is not subject to specific legal or regulatory requirements, including HIPAA, or IRB certifications. The HAM10000 dataset is free from licensing or authorization requirements.

Initially Google Drive was used for data storage. The data is accessed only by authorized team members. After augmentation if the dataset size increases more than 8GB, utilize buckets on GCP (Google Cloud Platform) which provides security and scalability. IAM is implemented for buckets in GCP for secure and controlled access.

Project Deliverables

The project deliverables include fully developed deep learning models capable of accurately classifying different classes of skin cancer and cleaned images dataset with hair removed. Continuous progress and improvements will be documented. Furthermore, a clear timeline of the tasks to be performed will be presented visually using Gantt and PERT charts.

Report

A comprehensive analysis of our project begins with a brief introduction, including the background on skin cancer prevalence, the importance of early detection, and the role of deep learning in skin lesion classification. The report includes a summary of findings from relevant research papers, detailing various approaches, datasets, and models used in skin lesion

classification, as well as their effectiveness, which can support the rationale for our chosen methodology. The data and project management will outline strategies for managing data, project development methodology, project organization, resource needs, and project timeline, utilizing tools such as Gantt and PERT charts.

In the section of data engineering, the report will cover data collection of HAM10000 and the reason behind choosing it, data pre-processing techniques utilized, transformation from raw dermatological images into high-quality images, appropriate methodologies applied to partition dataset into training, validation, and testing datasets. Additionally, visual representations of the evolving outcomes will be provided through statistical analysis, followed by the utilization of big data analytical techniques for data analytics. In the section dedicated to model development, the report will outline the selected deep learning techniques employed for model development. It will offer comprehensive explanations to enhance understanding of the models, elaborating on the reasoning behind the selection of each model for deployment, considering its performance, strengths, weaknesses, and limitations. This section will culminate with the evaluation phase, distinctly specifying the metrics utilized for accuracy calculation, and visually displaying both the original and predicted images. In the end, the report summarizes the valuable insights and solutions derived from the project analysis, lessons learnt from the experience, and detailing the impact of the project on the skin cancer diagnosis, including the references, appendices if any additional supporting materials utilized for the project.

Prototype

The initial version of the "Skin Cancer Classification Using Deep Learning" project presents the deep learning models, capable of categorizing skin lesions into different types of skin cancer. The groundwork involves preprocessing the HAM10000 dataset, encompassing tasks like resizing images to a standard size, and normalizing pixel values. Following preprocessing, the dataset will be divided

into three subsets: training, validation, and test sets. Model implementation will be carried out using deep learning frameworks like TensorFlow or PyTorch, facilitating the construction and training of complex neural network architectures such as convolutional neural networks (CNNs), CNN – One vs All (OVA), DenseNet121, and ResNet50.

Once implemented, the models will be trained using the training data to learn the mapping between input images and their corresponding class labels. The testing dataset will now be evaluated on the generated model to compare the performance of each of the deep learning models. This prototype will be analyzed deeply to understand where the models are lacking and what their strengths are. Based on the analysis, the models will be refined to achieve optimal accuracy utilizing different approaches. Through these comprehensive steps and considerations, the prototype aims to establish a robust foundation for an accurate skin cancer classification system.

Development Applications

The analysis on the initial project prototype will serve as the foundation for the further developments on the models. To enhance accuracy, the images will be pre-processed utilizing methods such as rotating, resizing, flipping, and grey scaling. If the initial prototype lacks data for any certain class, data augmentation techniques like zooming, flipping, and rotation will be used to address the imbalance by generating images. Furthermore, extensive hyperparameter tuning will be conducted to optimize model performance, considering factors like learning rate, regularization strength and batch size. Techniques such as grid search or random search will be employed to systematically explore the hyperparameter space and identify optimal configurations for improved performance. The developed model will now be evaluated on the test dataset using evaluation metrics F1 Score, AUC score and confusion matrix. The visualization of actual and interpreted images will further define the model's development.

Production Applications

The Skin Cancer Classification Using Deep Learning project has significant implications for healthcare applications and more. Once fully developed, these deep learning models could be integrated into current clinical systems to help dermatologists and medical professionals accurately identify skin lesions. Utilizing various preprocessing methods and strong model designs, the system aims to provide dependable and efficient classification of various skin cancer types, aiding in early detection. The evaluation of the model's effectiveness using real-time data will begin after its deployment on a web platform. Here, patient information will be gathered adhering to the privacy regulations and used to test the models, seeking feedback from medical experts. As the system continues to evolve through ongoing refinement and optimization, it promises to significantly impact the field of dermatology by enhancing diagnostic accuracy, improving patient outcomes, and ultimately saving lives.

Work Breakdown Structure

The skin cancer classification project follows the six-stage CRISP-DM methodology. Initially, in the business understanding phase, we set up the project team, establish communication, review relevant literature, and evaluate resources and potential challenges. Following that, the data understanding phase involves an understanding of the HAM10000 dataset to identify lesion characteristics and evaluate the quality of data, resulting in a comprehensive report. In the data preparation stage, we engage in activities like configuring the GCP cloud, applying k-means clustering with local binary patterns and colour percentiles, expanding the dataset using data augmentation techniques like zooming, flipping, rotation, and standardizing the data. The modelling phase is where we develop initial classification models, finetune them, train with the augmented dataset, and refine them by adjusting hyperparameters. The evaluation stage is focused on assessing the performance of these models and enhancing them for more precise lesion classification, with

detailed documentation of the results. Finally, the deployment stage involves putting the models into live use, conducting real-time testing, and preparing a detailed final report and presentation to conclude the project.

Gantt & PERT Charts

Gantt and PERT charts, along with the Work Breakdown Structure (WBS), are key visual aids in project management. Gantt charts offer a detailed timeline that shows how long tasks will take, their interdependencies, and deadlines, making it easier to keep track of the project's progress. PERT charts, on the other hand, are network diagrams that outline various tasks and their estimated durations, emphasizing the crucial path to enhance workflow efficiency. Used together, these tools form a comprehensive framework that helps in the successful completion of the project. Designed these tools to make the project flow, visually understandable. Table 1 summarizes the timeline of deliverables.

Table 1

Timeline of the Deliverables

Deliverable	Description	Due Date
Project Proposal	Documentation of research problem statements, and the detailed approach.	02/16/2024
WBS	WBS details each task assigned to every team member.	03/01/2024
Gantt Chart and PERT Chart	Presents the progress across the project with a timeline of the tasks.	03/22/2024
Data and Project Management Plan	Outline strategies for managing data, project development methodology, project organization, resource needs, and project timeline.	04/05/2024

Deliverable	Description	Due Date
Data Collection Plan	Understanding the HAM10000 dataset.	04/12/2024
Data Engineering Plan	Data pre-processing techniques, followed by partitioning the datasets, alongside visualizing evolving outcomes through statistical analysis, and applying big data analytical techniques.	04/19/2024
Presentation Slides	Presentation of methodology, tools, literature, and technologies used.	05/10/2024
Project Report	Documentation of the project from problem statement to impact of the application.	05/17/2024

Technology and Solution Survey

As mentioned above, skin cancer is one of the most deadly and widespread forms of cancer in the world and if detected early, the spread could be stopped. The traditional diagnostic methods which are done physically by the medical professionals are very time consuming and often subjective. However, the advent of computer-aided diagnostics in the early 1980s, marked a significant step in efficient skin cancer detection methods. Furthermore, with the recent technological advancements in the field of dermatological imaging and deep learning technologies skin cancer classification has been revolutionized. Investigating the various current technologies utilized in the classification of skin cancer lesions is crucial for advancing our project and harnessing these innovations for better diagnostic solutions.

Syafa'ah et al. (2021) explores the application of Convolutional Neural Networks (CNNs) for classifying skin lesions into seven categories, including melanoma and non-melanocytic lesions. The paper highlights a common challenge which is the lack of sufficient training data. To tackle this, the paper implemented data augmentation techniques, including rotating, zooming, and altering the

dimensions – height and width of the images in the dataset. This enabled the CNN model to generalize and better understand and interpret the data. The paper talks about three testing categories, melanocytic and non- melanocytic classes, melanoma, and melanocytic nevus class using the melanoma data, and finally malignant and benign class. The paper demonstrates how more data got by data augmentation will result in improved accuracy of the CNN model for skin cancer lesions classification.

Gururaj et al. (2023) talks about the effectiveness of CNN in classification problems. They employed transfer learning CNN models DenseNet169 and ResNet50 to classify skin cancer lesions. A major focus of the research is on data preprocessing techniques, such as the Dull Razor method for hair removal in the images and segmentation of images using an encoder-decoder architecture, which are crucial for enhancing the quality of dermatoscopic images before they are fed into the models. They also employed sampling techniques to balance the dataset so that the classes are more evenly distributed. These preprocessing steps proved to be very in facilitating a more accurate classification. This approach underscores the potential of combining advanced CNN architectures with targeted data preprocessing to improve early skin cancer classification, offering a significant contribution to the field, and setting a new benchmark for future research.

Saeed and Zeebaree (2021) explore the application of Convolutional Neural Networks (CNNs), for the classification of skin cancer from dermatoscopic images. The paper also discusses the ABCDE rule, which is a clinical method used by dermatologists to evaluate the characteristics of skin lesions for melanoma. The acronym stands for Asymmetry, Border irregularity, Color variation, Diameter greater than 6mm, and Evolution over time. However, the ABCDE rule is prone to be highly subjective and may not always accurately distinguish between benign and malignant lesions. This highlights the need for more objective and reliable diagnostic tools, such as deep learning models. The research also introduces a hybrid model approach that combines both traditional machine

learning techniques and deep learning techniques. This hybrid model aims to improve skin cancer detection by leveraging the strengths of both methodologies. This integrated model combines features identified through traditional analysis and deep-Convolutional features. In all, the paper highlights how deep learning techniques, particularly CNNs, and hybrid models can enhance the accuracy and reliability of skin cancer classification, providing a promising direction for future research.

Harangi et al. (2020) introduces a novel approach to enhance the classification of skin cancer lesion images into seven distinct classes by using the binary classification support. This approach makes use of the fact that skin cancer can be effectively classified into two broader categories, healthy and diseased. And the network model was first trained on this binary classification. The confidence levels generated from this binary classification were then utilized to adjust the multi-class confidence values, thus enhancing the overall performance of the network. The GoogLeNet Inception-v3 model was used for both binary and multi-class classification tasks. The innovation of integration of binary classification confidence into multi-class confidence values significantly boosted the overall accuracy of the multi-class classification. The research also emphasizes the importance of data preprocessing techniques, such as noise reduction and image augmentation, to address the challenge of insufficient training data and to balance the dataset, so that the classes are more evenly distributed. This innovative approach presented in the paper, which combines the strengths of both binary and multi-class classification, is a significant advancement in the classification of skin cancer lesions.

Harangi et al. (2018) presents an innovative approach for skin cancer classification by using a combination of Convolutional Neural Networks (CNNs) – specifically which won in the worldwide Image Classification Challenge ImageNet - AlexNet, VGGNet, and GoogLeNet. The ensemble method integrates the strengths of each individual CNN architecture to improve the overall classification

accuracy of skin cancer lesions. By aggregating the weighted outputs of each CNNs into a single neural network architecture, the ensemble facilitates simultaneous parameter optimization through backpropagation during training. The paper speaks about addressing the issue of overfitting due to combined architecture can be addressed by regularization and by increasing the volume of the training data. This method not only enhances classification performance but also allows for the ensemble to be trained as a unified system. The results indicate that the ensemble model outperforms the individual performances of its member CNNs, thus underlining the effectiveness of combining multiple deep learning models for skin cancer classification and paving way for further research in ensemble models.

Rashid et al. (2019) talks about not “leveraging the true potential of deep learning” in the field of medical image classification due to limited dataset sizes available. To overcome this, the paper uses Generative Adversarial Networks (GANs) to generate realistic looking dermatoscopic images. A GAN operates by training two networks simultaneously - The Generator network, generates fake images, by learning from the underlying data distribution, and the Discriminator network estimates whether the fake image generated by generator network is authentic or not. Through this approach of augmenting the training dataset for deep convolutional neural networks, such as DenseNet and ResNet architectures, the authors demonstrate that there has been a significant performance improvement in skin cancer lesion classification compared to conventional data augmentation strategies. This is a promising direction for future research in data generation and augmentation techniques, where large datasets are not available.

Polat and Koc (2020) presented two methods for skin cancer lesion classification: a standalone Convolutional Neural Network (CNN) model and a combination of CNN with a one-versus-all (CNN-OVA) strategy. The CNN-OVA method employs seven different two-class models, each differentiating one specific class from all others combined, finally taking the average of all such

pairwise results to classify the image. This OVA approach significantly enhances the classification performance, achieving an accuracy greater than the standalone CNN model's accuracy. Thus, the use of OVA, alongside CNNs, offers a promising way for improving diagnostic performance in image classification paving a solid path for future research and innovative ways in improving the classification performance.

Zhao (2022) evaluates and compares two different deep learning methods - Convolutional Neural Networks (CNNs) and Transformers in classifying skin cancer lesions. To do this comparison the paper employs two CNN architectures, VGGNet and ResNet, and two transformer-based models, Vision Transformers (ViT) and DeepViT. The paper also addresses the issue of dataset imbalance and the size of the training set. Strategies such as assigning different weights to lesions, image cropping, and data augmentation through methods like random horizontal flips and rotations were employed. The study concludes that both architectures perform well. And of the two, CNN models (VGGNet and ResNet) demonstrate higher performance compared to the transformer methods (ViT and DeepViT), with ResNet achieving the highest accuracy among the models tested. The paper concludes that CNNs provide a more reliable approach for skin cancer classification and suggests combining the strengths of CNN models and Transformers for enhanced performance in skin cancer classification.

Wu et al. (2022) employs deep convolutional neural networks (CNNs) using transfer learning for skin cancer lesion classification. The paper tailors the transfer learning models - InceptionV3, ResNet50, and DenseNet201 by modifying their structures—removing output layers, adding new pooling and fully connected layers, and integrating some original convolution and pooling layers with these new components. These modified models are then fine-tuned with parameters from the ImageNet network to achieve classification. To ensure the models are trained on high-quality, balanced datasets and to improve the performance of the classification, the authors employed

various data preprocessing techniques such as image cropping to focus on lesion areas, removal of hair to reduce visual interference, and application of sampling techniques to address the issue of dataset imbalance. Of all the models employed, ResNet50 proved to be more efficient, and the paper suggests that transfer learning, with tailored CNN architectures, significantly enhances the capability to accurately classify skin lesions. This paves a path for future research in adapting and enhancing pre-existing models for various applications.

Neeshma and Nair (2022) used DenseNet-121, a pretrained deep learning architecture for skin cancer classification by focusing the research on the impact of the balancing nature of the dataset on the classification performance through two experiments: one using the original imbalanced dataset and another with a resampled balanced dataset. Apart from balancing the dataset no other form of data preprocessing or augmentation techniques were employed. The experiment shows a significant improvement in the classification accuracy when a balanced dataset is employed. This enhancement underscores the importance of balanced datasets and paves a way for future research in more advanced data augmentation techniques and generative models to make the datasets more balanced.

Li and Seo (2021) talk about the detection and classification of skin cancer using deep learning techniques, through the application of Mask Region-based Convolutional Neural Network (Mask RCNN) models enhanced by transfer learning. To address the issue of data availability and the complexity of the dermatoscopic images – various shapes/sizes, the uses pre-trained weights from the Microsoft COCO dataset to train the Mask R-CNN model for skin lesion segmentation. This model's learned weights are then subsequently transferred and applied to the classification task, effectively making use of the insights gained from segmentation to improve classification performance. The methodology includes significant image preprocessing steps such as cropping, resizing, color correction, contrast adjustment, hair removal, and denoising to ensure the quality of

images fed into the models. This study presents a promising direction for future research by suggesting that further advancements could be made by exploring more sophisticated data augmentation techniques, leveraging larger datasets, and optimizing model architectures and training parameters to handle class imbalances more effectively.

Table 2 shows the comparisons of various models and their research.

Table 2

Comparison of Solutions

Author/Year	Model	Approaches	Key Findings and Conclusions	Results
Lailis Syafa'ah et al., 2021	CNNs	Three testing scenarios	melanoma and melanocytic nevus class test scenario has the highest accuracy	94% accuracy
Gururaj et al., 2023	DenseNet169, ResNet50	Dull Razor method and encoder-decoder architecture for enhancing the data Hybrid model	Targeted data preprocessing improved skin cancer classification performance Hybrid models enhance the performance of classification	0.912 AUC score
Saeed et al., 2021	CNNs	traditional machine learning and deep learning binary classification supporting multiclass classification	Performance can be improved by integrating binary classification outcome significant performance improvement was observed by enhancing the training dataset via GAN produced images	The accuracy raised by 7% by integrating binary classification
Harangi et al., 2020	GoogLeNet Inception-v3			
Rashid et al., 2019	DenseNet and ResNet and GANs	Generate realistic looking images via GAN model		86.1% accuracy

Author/Year	Model	Approaches	Key Findings and Conclusions	Results
Polat et al., 2020	CNN and CNN OVA	CNN-OVA combines seven different two-class models	OVA approach significantly enhances the performance	CNN 77% and CNN-OVA 92.9%
Zhao, 2022	CNNs - VGGNet and ResNet and Transformers - ViT and DeepViT	Data augmentation image cropping, rotations.	CNN models demonstrated higher performance when compared with Transformers.	VGG16 93.77% ResNet18 94.38% ViT 84.08% DeepViT 85.59%
Wu et al., 2022	Transfer Learning models - InceptionV3, ResNet50, and DenseNet201	Modified their architectures and are fine-tuned with the parameters of ImageNet	Of all the models, ResNet50 is seen to be more efficient	ResNet 50 86.69%

Literature Survey of Existing Research

(2021) delve into the utilization of deep learning methodologies, particularly focusing on Convolutional Neural Networks (CNNs) and Artificial Neural Networks (ANNs), for the detection of skin cancer. The paper extensively covers the processes of data preprocessing and feature extraction, meticulously detailing the architecture of the employed neural networks. By evaluating the diagnostic accuracy of these models, the study not only underscores the potential applications within the healthcare sector but also navigates through the various challenges encountered in the process. Offering significant contributions to the field of dermatological diagnostics, this research highlights the importance of advancing these computational methods to enhance the effectiveness of skin cancer detection. It suggests avenues for future research aimed at refining diagnostic tools, emphasizing the critical need for diverse and robust datasets.

to improve the training and performance of such models. Through this exploration, Dildar et al. (2021) presents a compelling narrative on the transformative potential of deep learning in revolutionizing skin cancer diagnostics, advocating for a multidisciplinary approach to develop more sophisticated and accurate diagnostic technologies.

In their study, Sheng et al. (2019) investigates the efficacy of high-resolution MRI with microscopy coil in preoperatively evaluating facial nonmelanoma skin cancers, focusing on the tumours' extent, depth, margins, and their relationship with adjacent structures. The prospective study involved 15 patients with 16 lesions, comparing HR-MRI findings with intraoperative observations and histopathological results, particularly the depth of invasion. The research found HR-MRI to be a precise, noninvasive tool, correlating strongly with histopathological findings and aiding in optimal surgical planning by accurately predicting tumour characteristics, thereby minimising the need for additional excisions and preventing recurrences during the follow-up period. This study underscores the significant role of advanced imaging techniques in enhancing the management and treatment outcomes of facial nonmelanoma skin cancers.

Moreno-Ramirez et al. (2015) explored the application of telemedicine in dermatology, particularly its role in managing skin cancer cases. They investigated the integration of various technologies such as teledermatology platforms, image analysis software, and teleconsultation tools, often incorporating them into existing electronic health record systems. Through methods ranging from remote consultations to real-time video conferencing and machine learning algorithms for image analysis, they demonstrated the efficacy of telemedicine in achieving high diagnostic concordance rates comparable to traditional face-to-face consultations. Despite challenges such as data security concerns and limitations in virtual examinations, their research underscored the importance of telemedicine in reducing healthcare disparities and improving patient outcomes, calling for its integration into routine clinical practice to realise its full potential.

Looking ahead, future research in this field should focus on addressing technical challenges, optimizing algorithms for automated diagnosis, and developing standardised protocols for the implementation of telemedicine in dermatology practice. By harnessing the potential of telemedicine, particularly in leveraging artificial intelligence and deeplearning techniques for image analysis, healthcare systems can enhance access to care, reduce diagnostic delays, and improve outcomes for skin cancer patients while ensuring the security andquality of remote consultations.

Fujisawa et al. (2020) embarked on apioneering exploration of deep learning-based computer-aided classifiers for skin tumour diagnosis, utilising a small dataset of clinical images. Their study showcased the development ofclassifiers that surpassed the diagnostic accuracy of board-certified dermatologists. Leveraging deep learning techniques, they trained models to analyse clinical images and classify skin tumours with remarkable precision. By doing so, they demonstrated the potential of artificial intelligence in augmenting dermatologists' diagnostic capabilities, particularly in scenarios with limited access to expert consultation or resources.Despite the groundbreaking achievements, Fujisawa et al. encountered challenges inherent to working with small datasets, including issues related to generalisation and robustness. However, their work underscored the importance of continued research and development in deep learning approaches for skin tumour diagnosis, withimplications for improving access to accurate and timely dermatological care. Moving forward, future studies should focus on expanding dataset sizes, refining model architectures, and validating classifiers in diverse clinical settings to enhance their applicability and reliability in real-world practice.

In their publication, Zhang et al. (2018) introduced a novel approach to automatic skin lesion segmentation by combining deep fully convolutional networks with shallow networks utilising textons. Their method aimed to capitalise on the strengths of deep learning for feature

extraction and segmentation while integrating textons to enhance segmentation accuracy by capturing finer details. Through this coupling of deep and shallow networks, they achieved robust segmentation results, surpassing conventional methods and advancing the state-of-the-art in skin lesion analysis. Their work showcased the potential of integrating deep learning with texture-based methods for more precise skin lesion segmentation, offering promising prospects for improving diagnostic accuracy in dermatology. However, challenges such as dataset variability and computational complexity may pose limitations to the scalability of the proposed approach. Nonetheless, Zhang, Yang, and Ye's research underscores the significance of combining advanced computational techniques with traditional texture-based methods to enhance skin lesion segmentation accuracy. Moving forward, further refinements in model architectures and extensive validation across diverse datasets and clinical scenarios are essential to establish the efficacy and applicability of this approach in real-world dermatological practice.

Kasuya et al. (2017) research marks a significant advancement in dermatological imaging by offering a unique perspective on dermoscopic structures. Through digitally reconstructed pathological horizontal top-down view images, they provided a clearer and more intuitive representation of dermoscopic findings, facilitating easier interpretation and analysis for dermatologists and clinicians. By presenting dermoscopic structures in a format akin to the view from above a skin lesion, this method enhances the understanding of complex dermatological conditions and aids in accurate diagnosis. Furthermore, the approach holds promise for improving medical education and training in dermatology, potentially leading to better patient care and outcomes.

In their comparative analysis, Pellacani et al. (2016) meticulously scrutinized the morphological parameters extracted from pigmented skin lesion images acquired via epiluminescence surface microscopy (ELSM) and polarized-light videomicroscopy (PLVM). By

systematically assessing the imaging characteristics and the ability of each modality to capture key morphological features, the researchers aimed to elucidate potential disparities in diagnostic efficacy between the two techniques. Their findings offer critical insights into the nuanced differences in lesion visualization and characterization, empowering clinicians to make informed decisions regarding the selection of imaging modalities tailored to individual patient cases. Additionally, this research lays the groundwork for further advancements in dermatological imaging technology, potentially leading to the refinement of diagnostic protocols and the development of novel techniques to enhance the accuracy and reliability of pigmented lesion assessment. Pellacani and Seidenari's investigation underscores the importance of methodical evaluation and comparison of imaging modalities in dermatology, particularly concerning the diagnosis and management of pigmented skin lesions. By elucidating the distinct advantages and limitations of epiluminescence surface microscopy (ELSM) and polarized-light videomicroscopy (PLVM), their study provides valuable guidance for clinicians seeking to optimize diagnostic strategies for pigmented lesion assessment. Furthermore, their findings may inform future research endeavors aimed at refining existing imaging technologies or developing innovative approaches to enhance lesion visualization and characterization. Ultimately, this research has the potential to improve clinical outcomes and patient care by facilitating more accurate and timely diagnosis of pigmented skin lesions, thereby contributing to advancements in dermatological practice.

In their survey, Ali et al. (2020) focused on the automation of the ABCD rule for melanoma detection, which constitutes a pivotal step in dermatological diagnosis. By harnessing advancements in computer vision and machine learning, the researchers sought to streamline the identification process of potentially malignant skin lesions. Their survey likely explored a plethora of computational approaches and algorithms tailored to analyze dermatoscopic

images, extract pertinent features, and evaluate them against the ABCD criteria. This automation endeavor holds significant promise in not only expediting the detection of melanoma but also potentially improving diagnostic accuracy, thereby fostering early intervention strategies and ultimately enhancing patient outcomes in the realm of skin cancer management. Automating the ABCD rule for melanoma detection represents a significant advancement in dermatological imaging and diagnostic practices. Ali, Li, and Yang's survey sheds light on the integration of computational techniques into traditional diagnostic paradigms, offering insights into how machine learning algorithms can augment the capabilities of dermatologists in identifying suspicious skin lesions. By systematically examining the landscape of automated approaches to melanoma detection, their work paves the way for the development of robust and efficient tools that could revolutionize the field of dermatology, enabling earlier detection, more accurate diagnosis, and improved patient care.

In their study, Henning et al. (2019) introduced the CASH (Color, Architecture, Symmetry, and Homogeneity) algorithm as a novel approach for dermoscopy, a crucial tool in dermatological diagnostics. This algorithm aimed to enhance the assessment of skin lesions by incorporating four key criteria: color, architecture, symmetry, and homogeneity. By systematically evaluating these parameters, the CASH algorithm provided dermatologists with a structured framework for analyzing dermoscopic images and identifying suspicious lesions. This approach represents a significant advancement in dermoscopy, offering a comprehensive and standardized method for lesion evaluation that can potentially improve diagnostic accuracy and streamline clinical decision-making processes. Moreover, the introduction of the CASH algorithm by Henning et al. underscores the ongoing efforts to develop innovative computational tools to aid in the early synthesis and to predict cancer subtype, grade, stage, and patient prognosis. Moreover, Kuntz, Krieghoff-Henning, Kather, Jutzi, Höhn, Kiehl, and their colleagues' systematic review sheds

light on the challenges, opportunities, and future directions in utilizing deep learning for gastrointestinal cancer classification and prognostication. By synthesizing the current state-of-the-art methodologies and research findings, their review offers valuable insights for clinicians, researchers, and developers working in oncology and computational pathology. Furthermore, their work likely provides a roadmap for the development and validation of robust deep learning models that can aid in clinical decision-making, treatment planning, and patient management in gastrointestinal cancer. Ultimately, the systematic review detection of skin cancer and other dermatological conditions. By leveraging advanced image analysis techniques and machine learning algorithms, the CASH algorithm holds promise in augmenting the capabilities of dermatologists and enhancing the efficiency of skin lesion evaluation. The integration of such algorithms into clinical practice has the potential to revolutionize dermatological diagnostics, leading to earlier detection, improved patient outcomes, and ultimately, a reduction in the burden of skin cancer worldwide.

In their examination, Perez et al. (2018) delved into the critical decision-making process concerning the selection of either solo or ensemble Convolutional Neural Network (CNN) architectures for melanoma classification. With the increasing complexity and variability of dermatological images, the choice between these architectures becomes pivotal in achieving accurate and reliable classification results. Their study likely explored the strengths and limitations of both solo CNNs, which operate independently, and ensemble CNNs, which combine multiple models to enhance performance. By scrutinizing factors such as computational efficiency, robustness, and classification accuracy, Perez, Avila, and Valle provided valuable insights into the optimal approach for melanoma classification using CNNs. Their research contributes to advancing the field of dermatological imaging by elucidating the most effective strategies for leveraging deep learning techniques in skin cancer diagnosis, potentially leading to improved

diagnostic accuracy and patient outcomes. Furthermore, Perez, Avila, and Valle's investigation underscores the growing importance of leveraging advanced computational techniques to address the challenges inherent in melanoma classification. By systematically evaluating the performance of solo and ensemble CNN architectures, their study sheds light on the nuances of model selection and optimization in dermatological image analysis. This research not only enhances our understanding of the capabilities of CNNs in melanoma classification but also provides valuable guidance for clinicians and researchers seeking to implement machine learning algorithms in clinical practice. Ultimately, the insights gleaned from their examination have the potential to catalyze the development of more robust and efficient diagnostic tools for skin cancer detection, leading to improved patient care and outcomes.

In their review, Li et al. (2020) centered their attention on the application of deep learning techniques for skin disease diagnosis, marking a significant advancement in dermatological imaging and diagnostic practices. By synthesizing existing literature and research findings, their review provided a comprehensive overview of the current landscape of deep learning methodologies in the field of dermatology. Deep learning techniques, characterized by their ability to automatically learn complex patterns and features from large datasets, offer immense potential in improving the accuracy and efficiency of skin disease diagnosis. Li et al.'s review likely explored various deep learning architectures, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and their variants, and their applications in classifying and diagnosing a wide range of skin diseases. Furthermore, Li, Pan, Zhao, and Zhang's review highlights the transformative impact of deep learning techniques on dermatological diagnostics, offering insights into the strengths and limitations of these approaches. By systematically evaluating the performance of deep learning models in skin disease diagnosis, their review contributes to advancing our understanding of the capabilities of artificial intelligence in

dermatology. Moreover, their work likely provides valuable guidance for researchers, clinicians, and developers seeking to harness the power of deep learning for improving patient care and outcomes in dermatological practice. Ultimately, the insights gleaned from their review have the potential to drive innovation and foster the development of more accurate, efficient, and accessible diagnostic tools for skin diseases.

In their comprehensive survey, Hameed et al. (2017) undertook a meticulous examination of image-based computer-aided diagnosis (CAD) systems tailored specifically for skin cancer. By the broad array of research literature and advancements in the field, their survey aimed to provide a holistic understanding of the current landscape of CAD systems for skin cancer diagnosis. Image-based CAD systems play a pivotal role in enhancing diagnostic accuracy by leveraging advanced computational techniques to analyze dermatological images and detect suspicious lesions. Hameed et al.'s survey likely covered a wide range of CAD methodologies, including machine learning algorithms, deep learning architectures, and feature extraction techniques, and their applications in diagnosing various types of skin cancer. Furthermore, Hameed, Ruskin, Hassan, and Hossain's survey shed light on the challenges, opportunities, and future directions in image-based CAD systems for skin cancer diagnosis. By systematically evaluating the performance, limitations, and potential advancements of existing CAD systems, their survey offers valuable insights for researchers, clinicians, and developers working in the field of dermatology. Moreover, their work likely provides a roadmap for the development of more accurate, efficient, and accessible CAD systems that have the potential to revolutionize skin cancer diagnosis and management. Ultimately, the comprehensive survey conducted by Hameed et al. contributes to advancing the state-of-the-art in dermatological imaging and diagnostic practices, paving the way for improved patient outcomes and enhanced healthcare delivery.

In their systematic review, Kuntz et al. (2018) conducted a comprehensive analysis of

gastrointestinal cancer classification and prognostication from histology utilizing deep learning methodologies. Their review aimed to consolidate the existing body of literature and advancements in the field, focusing on the application of deep learning techniques to analyze histopathological images for accurate cancer classification and prognostication. Histological analysis plays a crucial role in cancer diagnosis and prognosis, and deep learning offers immense potential in automating and enhancing this process. Kuntz et al.'s review likely explored various deep learning architectures, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), and their applications in analyzing histological images conducted by Kuntz et al. contributes to advancing the field of computational pathology and oncology, facilitating more accurate and personalized cancer care for patients.

In their review, Pathan et al. (2019) provided a comprehensive overview of techniques and algorithms employed in computer-aided diagnosis (CAD) systems for pigmented skin lesions. This systematic examination aimed to consolidate the diverse methodologies and advancements in the field, focusing specifically on enhancing the diagnostic process for pigmented skin lesions through computational approaches. CAD systems play a crucial role in dermatological practice by leveraging advanced image analysis techniques and machine learning algorithms to assist dermatologists in diagnosing skin lesions accurately. Pathan et al.'s review likely explored a range of techniques, including feature extraction methods, segmentation algorithms, and classification models, tailored to analyze dermatoscopic images and aid in the detection and classification of pigmented skin lesions. Furthermore, Pathan, Prabhu, and Siddalingaswamy's review likely highlighted the challenges, opportunities, and future directions in CAD systems for pigmented skin lesion diagnosis. By synthesizing the existing literature and research findings, their review offers valuable insights for researchers, clinicians, and developers seeking to develop and refine CAD systems in dermatology. Moreover, their work provides guidance for the development of more

accurate, efficient, and accessible CAD systems that can aid dermatologists in diagnosing pigmented skin lesions with greater accuracy and confidence. Ultimately, the review conducted by Pathan et al. contributes to advancing the field of dermatological imaging and diagnostic practices, facilitating improved patient outcomes and healthcare delivery.

In their systematic review, Brinker et al. (2018) concentrated on skin cancer classification utilizing convolutional neural networks (CNNs), representing a significant advancement in dermatological imaging and diagnostic practices. CNNs, a type of deep learning architecture inspired by the organization of the animal visual cortex, have demonstrated remarkable capabilities in automatically learning hierarchical features from medical images, including dermatoscopic images of skin lesions. Brinker et al.'s review likely synthesized existing literature and research findings to evaluate the performance and efficacy of CNNs in accurately classifying various types of skin cancer lesions based on dermatoscopic images. This review explored the challenges, opportunities, and future directions in skin cancer classification using CNNs. By analyzing the current state-of-the-art methodologies and advancements in the field, their review offers valuable insights for clinicians, researchers, and developers working in dermatology and medical imaging.

Furthermore, their work likely provides guidance for the development and validation of robust CNN models tailored to skin cancer classification, potentially leading to more accurate and efficient diagnostic tools. Ultimately, this paper contributes to advancing the field of dermatological imaging and diagnostic practices, facilitating improved patient outcomes and healthcare delivery in the realm of skin cancer diagnosis and management. Comparison of research papers is summarized in the Table 3.

Table 3*Comparison of Relevant Research Papers*

Author/Year	Goal	Approach	Targeted Problem	Conclusion
Dildar et al., 2021	Skin cancer detection using deep learning techniques	Utilization of CNNs and ANNs for skin cancer detection, data preprocessing, and feature extraction	Emphasizing the critical need for diverse and robust datasets	Multidisciplinary approach for sophisticated diagnostic technologies
Sheng et al., 2019	Preoperative evaluation of facial nonmelanoma skin cancers	Utilization of high-resolution MRI with microscopy coil for evaluation	Enhancing surgical planning through precise imaging	Advanced imaging techniques improve treatment outcomes
Moreno-Ramirez D, Ferrandiz L., 2015	Telemedicine applications in dermatology, particularly in skin cancer cases	Integration of various technologies like tele dermatology platforms and machine learning algorithms	Addressing data security concerns and technical challenges	Telemedicine reduces healthcare disparities and improves outcomes
Fujisawa et al., 2020	Deep learning-based computer-aided classifiers for skin tumor diagnosis	Utilization of deep learning on a small dataset of clinical images	Overcoming diagnostic accuracy of dermatologists	Deep learning enhances diagnostic capabilities
Zhang et al., 2018	Automatic skin lesion segmentation	Combination of deep fully convolutional networks with shallow networks utilizing textons	Dataset variability and computational complexity	Integration of deep learning with texture-based methods

Author/Year	Goal	Approach	Targeted Problem	Conclusion
Kasuva et al., 2017	Digitally reconstructed pathological horizontal top down view images for intuitive explanation of dermoscopic structures	Digitally reconstructed images offer clearer and intuitive representation	Facilitating easier interpretation for dermatologists	Improved understanding and diagnosis of dermatological conditions
Pellacani G, Seidenari S., 2016	Comparison of imaging modalities in pigmented skin lesion visualization	Systematic comparison of ELSM and PLVM imaging modalities	Nuanced differences in lesion visualization	Empowering clinicians with informed decisions
Ali et al., 2020	Automation of the ABCD rule for melanoma detection	Leveraging advanced computational techniques for automated detection	Expediting detection of melanoma	Machine learning augments diagnostic capabilities
Henning et al., 2019	Introduction of the CASH algorithm for dermoscopy	Incorporation of color, architecture, symmetry, and homogeneity criteria	Enhancing lesion evaluation with a structured framework	Comprehensive and standardized method for lesion assessment

Data and Project Management Plan

Data Management Plan

Data Collection Approaches

The Research utilizes publicly made available open-source dataset, HAM10000 published by Harvard Dataverse. This enables reproducibility and utilization of the research's methodology and results and further improve it by research in future.

The HAM10000 (Human Against Machine) dataset is a substantial compilation of around 10,015 dermatoscopic images in JPEG format were collected from different populations for over a period of 20 years from two sites, the Department of Dermatology at the Medical University of Vienna, Austria, and the skin cancer practice of Cliff Rosendahl in Queensland, Australia (Tschandl P. et al. 2018). This dataset aims to serve as training images for developing deep learning models that can recognize and classify skin cancers from images. The dataset was downloaded from Harvard Dataverse (Tschandl 2018). The metadata of these images is illustrated in Figure 3.

Figure 3

HAM10000_metadata

1	lesion_id	image_id	dx	dx_type	age	sex	localization	dataset
2	HAM_0000118	ISIC_0027419	bkl	histo	80	male	scalp	vidir_modern
3	HAM_0000118	ISIC_0025030	bkl	histo	80	male	scalp	vidir_modern
4	HAM_0002730	ISIC_0026769	bkl	histo	80	male	scalp	vidir_modern
5	HAM_0002730	ISIC_0025661	bkl	histo	80	male	scalp	vidir_modern
6	HAM_0001466	ISIC_0031633	bkl	histo	75	male	ear	vidir_modern
7	HAM_0001466	ISIC_0027850	bkl	histo	75	male	ear	vidir_modern
8	HAM_0002761	ISIC_0029176	bkl	histo	60	male	face	vidir_modern
9	HAM_0002761	ISIC_0029068	bkl	histo	60	male	face	vidir_modern
10	HAM_0005132	ISIC_0025837	bkl	histo	70	female	back	vidir_modern

Note. The figure shows the fields of the HAM10000 metadata.

The detailed naming conventions of the image files provide metadata of each lesion image indicating the type of skin cancer (dx), the method by which the type of skin cancer is confirmed (dx_type), the age of the patient (age), gender (sex), the area of the skin from where the lesion image was captured (localization) and the dataset. The skin cancer types included in the dataset are Actinic keratoses and intraepithelial carcinoma / Bowen's disease (akiec), basal cell carcinoma (bcc), benign keratosis-like lesions (bkl), dermatofibroma (df), melanoma (mel), melanocytic nevi (nv) and vascular lesions (angiomas, angiokeratomas, pyogenic granulomas and hemorrhage, vasc), which are confirmed through either histopathology (histo), or follow-up examination (follow_up), or expert consensus (consensus), or confirmation by in-vivo confocal microscopy (confocal) and the localization specifies from which part of the body is the lesion

image taken from like chest, face, back, neck and etc. Finally, the image_id is the filename of the image. Images filenames are represented as for e.g., ISIC_0024306 as shown in Figure 4.

Figure 4

Image Files



Note. The figure shows the dermatoscopic images of the HAM10000 file

The summary of key features of dataset is explained in Table 4.

Table 4

Summary of key Features of Dataset

Dataset	Modality	Total number of Images
HAM10000	Image	10015

Data Management Methods

In order to achieve an efficient and productive development cycle, the project used a structured data management approach with the CRISP-DM methodology and Agile Scrum

methodologies (Cao et al., 2014; Haq & Jackson, 2010; Schröer et al., 2018). Integrating this strategy across all project stages supported continuous change and improvement through a flexible and desirable project environment.

It was necessary to comply with the highest moral and legal obligations. The project's compliance objectives included extensive documentation of the origin of the HAM10000 dataset, ensuring that all data management processes followed copyright and intellectual property rights (IPR) standards. This commitment extended to obtaining informed consent in the right place, which emphasized the ethical integrity of the work.

The project was based on a well-planned, four-stage approach to data management. Each process, from data collection to metadata management to data storage and integration, is assigned to a specific team member, ensuring that every aspect of data management receives special attention. A detailed analysis of the metadata was necessary to understand the structure of the HAM10000 data system while adhering to stringent legal and ethical requirements.

While the HAM10000 dataset's open-source nature rendered it more accessible, the team recognized and acknowledged the original data providers' ownership rights. The emphasis on existing, ethically acquired datasets demonstrated the project's commitment to responsible data usage, since it excluded the collecting of additional personal or highly sensitive data.

Comprehensive documentation, including strict naming rules, secured the data's future interpretability and reliability. A strong storage and backup strategy, based on Google Cloud Platform (GCP) buckets, ensured data integrity and availability, ensuring the dataset's maintaining usage throughout the project.

The development process was organized into 1-2 weeks depending on the epics. Sprints within larger Stories and Epics, demonstrating the project's integrated approach with each stage assigned to dedicated team members which can be illustrated in Table 5. This systematic division

into key stages allowed for effective monitoring and for successful completion of project phases, with a designated coordinator managing the whole data management process and ensuring that project objectives were met while progressing smoothly.

Table 5

Details of the Data Management Stage

Data Management Phase	Team Member	Responsibilities
Data Acquisition	Harshitha Reddy Lingala	Focuses on acquiring the HAM10000 dataset, aligning it with research goals, and managing access for integration with computational resources.
Metadata Handling	Nikitha Goturi	Analyzes and documents dataset metadata, ensuring compliance with legal and ethical standards, facilitating its use in cloud environments.
Data Storage	Sai Kiran Baghavathula	Manages secure storage on GCP, leveraging cloud solutions for scalable data management, and oversees backup strategies.
Data Integration	Shivani Reddy Bakkannagari	Coordinates dataset integration within the project's platform, focusing on efficient processing and modeling in Google Colab.

Performed rigorous metadata analysis, secure storage in GCP, and evaluated data efficiency in Google Collab, with an emphasis on effective data governance, setting up access controls and mechanisms for maintaining a safe data environment. As responsibilities were carried out as research progressed, it became easier to develop reliable aggregation models for skin cancer classification.

Data Storage Methods

Data holds the utmost significance in a project, so ensuring its proper storage becomes crucial.

Throughout the project, Google Cloud Platform was utilized for efficient storage of the data. The dataset used in this project is ‘HAM10000’ dataset, which can be utilized for training, testing and validating the models built. This dataset is stored in the Google Cloud Platform in ham10000dataset bucket.

Figure 5

Google Cloud Buckets

The screenshot shows the Google Cloud Storage Buckets interface. At the top, there is a search bar with the placeholder "Search (/) for resources, docs, products, and more". Below the search bar are buttons for "CREATE" and "REFRESH". A notification banner at the top states: "Beginning on April 29th, 2024 at-scale policy analysis and advanced IAM recommendation capabilities will require Security Command Center Premium. [Learn more](#) [DISMISS]". The main area displays a table titled "Buckets" with one row. The columns are: Name, Created, Location type, Location, Default storage class, Last modified, and Public access. The single entry is "ham10000dataset" with values: Mar 28, 2024, 12:09:45 PM, Multi-region, us, Standard, May 9, 2024, 1:37:32 PM, and Not public.

Name	Created	Location type	Location	Default storage class	Last modified	Public access
ham10000dataset	Mar 28, 2024, 12:09:45 PM	Multi-region	us	Standard	May 9, 2024, 1:37:32 PM	Not public

Storing the dataset in bucket in GCP, enhances data security and privacy by providing control over access permissions and ensuring that users have only the access that is necessary for their role. GCP’s service, Google Cloud Storage allows the team to access shared resources from anywhere, ensuring consistency and accelerating project timelines. Since limited access ensures, the data is protected against threats and unauthorized access, maintaining the integrity and confidentiality of the sensitive information. Even if more data is to be added in the future, GCP’s scalable solution will dynamically adjust, providing a flexible environment for the data management.

Without reliable data storage, a project risks losing important data, which can lead to the loss of valuable insights and affect decision-making based on past information. Additionally, without centralizedand reliable data storage, the team might struggle to collaborate and share knowledge

effectively because members cannot access the same data or work together. Moreover, retaining data after completion of the project is crucial for enabling future improvements.

In conclusion, using Google Cloud Platform for data storage in project offers a secure, organized, and accessible way to handle dataset, essential for the project's success. By storing the data in GCP, ensure efficiency and security, enabling the team to collaborate effectively and adapt to any future data additions. Ethical standards and data principles were followed throughout the process.

Data Usage Mechanisms

Dataset was made accessible to every member of the team ensuring collaborative environment. Taken measures to guarantee that, even after the project concludes, the data remains accessible to those members who have the necessary permissions.

Selection and Preservation. HAM10000 dataset will be retained for future use since the dataset is open source and do not need additional charges to retain. Best practices were followed to secure the data, preserving data, ensuring that the data remains safe, reliable, and usable for a long time.

Data Sharing. Access to Google Cloud Storage, where the data is stored, is a key aspect of the data sharing strategy. Only authorized team members was granted access to this cloud storage, ensuring the data's security and controlled accessibility.

Figure 6*Access Control*

The screenshot shows the Google Cloud IAM Permissions page for the project "DATA270-GROUP3". The interface includes tabs for "PERMISSIONS" and "RECOMMENDATIONS HISTORY". A message at the top states: "Beginning on April 29th, 2024 at-scale policy analysis and advanced IAM recommendation capabilities will require Security Command Center Premium. [Learn more](#)" with a "DISMISS" button. Below this, the title "Permissions for project \"DATA270-GROUP3\" " is displayed, followed by a note: "These permissions affect this project and all of its resources. [Learn more](#)". There is a checkbox for "Include Google-provided role grants". The main table lists four principals:

Type	Principal	Name	Role	Security insights
<input type="checkbox"/>	harshithareddy.lingala@sjtu.edu	Harshitha Reddy Lingala	Editor	
<input type="checkbox"/>	nikitha.goturi@sjtu.edu	Nikitha Goturi	Owner	
<input type="checkbox"/>	saikiran.baghavathula@sjtu.edu	Sai Kiran Baghavathula	Editor	
<input type="checkbox"/>	shivanireddy.bakkannagari@sjtu.edu	Shivani Reddy Bakkannagari	Editor	

The project ensures to follow responsible data-sharing practices to maintain integrity and security of the data. This regulated access protects the information and makes remote and collaborative efforts possible. It allows team members to access and work on the data together, enhancing teamwork and ensuring data security at the same time.

Responsibilities and Resources, efficient data management was achieved by assigning specific team members responsibilities for different phases. Google Cloud Storage is utilized for data storage and seamless sharing.

Google Colab Pro was used for efficient data preprocessing, model building and evaluating. The project involved working with deep learning networks which utilizes advanced GPUs, and thus makes the Google Colab Pro the best choice.

Project Development Methodology

Analytics of Data with Intelligent Systems Development Cycle

This project makes use of a hybrid project development methodology, combining the Agile Scrum project development with the Cross-Industry Standard Process for Data Mining (CRISP-DM) as described by Schröer, Kruse and Marx Gómez (2021). This hybrid approach leverages the intelligent system development cycles of CRISP-DM for managing the data science lifecycle with Scrum's ability for continuous adaptation through feedback to enable a well-structured workflow. This hybrid approach ensures timely delivery of the project, with efficient tracking of the status of the deliverables and allowing for the scope of feedback and aligning the project with the evolving needs and requirements. The entire project overflow was divided into one-to-two-week sprint cycles depending on the epics and tasks that caters to a potentially project deliverable.

This hybrid project development methodology, combining the CRISP-DM and Agile Scrum, presents a comprehensive framework for navigating the process of data analytics. The sprint cycles provide for an agile and dynamic project development environment, which allows the Scrum Team to effectively address any challenges that arise in the project development and also allows them to incorporate feedback from the stakeholders and do the necessary changes before proceeding with the next deliverable. By incorporating Scrum Methodologies like Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective promotes transparency, accountability and teamwork among the Scrum Team which leads to effective project development. Finally, CRISP-DM and Agile Scrum Methodologies are used for this project's analytics-driven intelligent systems development cycle, to achieve the project's milestones in time and for the timely delivery of the entire project.

Planned development processes and activities.

The CRISP-DM Methodology consists of six phases that naturally describes the data science lifecycle. The six phases of CRISP-DM are: Business Understanding for determining project objectives and requirements, Data Understanding for exploring and collecting the required data, Data Preparation for organizing and formatting data for modeling, Modeling for building and assessing models to answer the business questions, Evaluation for determining the models' accuracy and to iterate further, and Deployment for integrating the final models either in real-time or in applications as per the requirements of the stakeholders. Through these six phases the business requirements are converted into solutions in a methodologically planned manner.

Business Understanding. This Skin Cancer Classification project seeks to leverage Artificial Intelligence to enhance the early detection and classification of skin cancer lesions by analyzing the dermatoscopic images of skin cancers. This project aims to train deep learning models, like Convolutional Neural Networks (CNNs), Transfer Learning models: ResNet50, DenseNet121, and Convolutional Neural Network - One-vs-All (CNN-OVA) model on these dermatoscopic images data to accurately classify skin cancer lesions. These automated classification systems are very important as the present manual diagnostic methods of detecting and classification of skin cancers are very subjective and time consuming, which can make the cancer spread. Models are evaluated against different metrics like F1 score, accuracy, recall, precision, and the best models are intended to be integrated in health infrastructure.

Data Understanding. In this phase of the project, careful exploration and identifying of dermatoscopic image dataset was done and HAM10000 dataset was considered for the project. This dataset was chosen for its diverse and comprehensive collection of skin lesion images, which was essential for training deep learning models to categorize skin cancers. The HAM10000 dataset consists of 10,015 dermatoscopic images categorizing skin lesion images into seven different

classes of skin cancer (Tschandl P. et al., 2018). The dataset was also analyzed to understand the distribution of skin cancer types, image quality, and to identify potential challenges for modeling like data imbalance or class imbalance, noise in the images. This understanding and the analysis of data, helped to make informed decisions for preprocessing the data in the data preparation phase.

Data Preparation. In the data preparation phase of the project, several meticulous steps are taken to make the data ready for modeling. They include cleaning the data, reducing the noise in the images like removal of hair to enhance the quality of the dermatoscopic images (Gururaj et al., 2023) and to tackle the data imbalance issue several data augmentation techniques are implemented including rotating, zooming and altering the dimensions of the images (Syafa'ah et al., 2021) and keras ImageDataGenerator was used to generate realistic looking dermatoscopic images to handle the issue of insufficient training data. This systematic and detailed approach to data preparation, emphasizing on the quality and the sufficiency of the data lays a strong foundation to the next phase in CRISP-DM methodology, which is modeling.

Modeling. In this phase several deep learning architectures, including the pre-trained models are explored and trained to detect and accurately classify the skin cancers. In this phase, various models including Convolutional Neural Networks (CNNs), Convolutional Neural Network - One-vs-All (CNN-OVA), pre-trained model architectures: ResNet50 and DenseNet121 are developed. CNNs are specialized for image analysis, building upon traditional neural networks to handle specifically image data, with many layers enabling them to learn complex patterns in the images (Fuadah et al., 2020). Then, CNN-OVA architecture was developed, which divides the multiclass classification into several binary classification models, thereby mitigating the issue of overfitting and accommodating training with fewer samples, thereby enhancing its generalization ability (Babu & Narayanan, 2022).

Another vertical of the modeling phase was training the pre-trained models: ResNet50 and DenseNet121. Residual Networks like ResNet50 (Polat & Güngen, 2021), trained on ImageNet dataset has the ability to mitigate the vanishing gradient problem and the transfer learning ability helps in accurately classifying skin cancer. The DenseNet121 architecture is a densely connected neural network, ensuring maximum information flow from the previous layer to its subsequent layers, thus making it ideal for learning complex patterns with fewer computational resources (Huang et al., 2016).

This phase also includes the hyperparameter tuning, which is a process to optimize the parameters of the neural network architectures like learning rate, number of hidden layers, number of neurons and the regularization parameter to yield the best model performance. This ensures that the models are efficient and generalized and not overfitting.

Evaluation. This is a very critical phase in the project development as it connects the theoretical aspects of the model development with the real-time, practical applications of the project. It aligns models and their performance to meet the real-time goals and the initial developed project goals in the business understanding phase.

In this phase the trained models of the modeling phase are evaluated against different metrics: F1 score, accuracy, specificity (recall), and sensitivity (precision). Accuracy provides the overall correctness of the models, while specificity and sensitivity throw light on the model's capability to correctly predict. The F1 score, harmonic mean of specificity and sensitivity, is vital in balancing the trade-off between the false negatives and false positives, which is a major challenge in medical diagnostics. The feedbacks from this phase are used to further refine the already trained models for better performance and the final models are made ready for the deployment.

Deployment. This is the final phase of the project, where the validated and the refined models are integrated in real-time or in applications as per the business requirements. This

integration of models into diagnostic platforms helps professionals in accurately categorizing skin cancers in no time.

The entire project including the project methodology, business understanding, data preprocessing steps, modeling architectures and evaluation results are to be well documented and be made available in the public domain for further collaborations and research to further improve the performance of categorizing the skin lesions.

Project Organization Plan

The Skin Cancer Classification using Deep Learning project follows the CRISP-DM methodology. Cross-Industry Standard Process for Data Mining (CRISP-DM) for Data Mining is a framework that outlines the stages of a data science project's life cycle. It provides a structured approach to planning, organizing, and executing the project, serving as a roadmap for the data mining life cycle (ProGlobalBusinessSolutions, 2023). This structured approach helps to break down complex activities of the project into simple tasks, making it possible to complete the project on planned timelines. Its iterative approach allows us to revisit the tasks whenever new insights are gained or feedback is collected. The CRISP-DM methodology is organized into six different phases, starting from Business Understanding to Deployment.

During the initial phase of CRISP-DM structure, known as Business Understanding, the project's objective and requirements are established and defined. This phase serves as the foundation for the skin cancer classification project utilizing deep learning, setting the stage for its subsequent phases. Following the Business Understanding phase, the Data Understanding phase allows for thorough examination of the HAM10000 dataset. This exploration helps in identifying necessary data transformations and aids in the selection of an appropriate deep learning model for subsequent stages.

The Data Preparation phase transforms the collected data in the GCP's buckets into a

form that is ready for modeling. This includes dealing with missing values, addressing outliers, and modifying the structure of the data into a format that is appropriate for analysis. Additionally, data standardization may be carried out to enhance the accuracy of the model. Essentially, this stage converts the raw data into a format that can be effectively used to train and test the models. In the modeling phase of CRISP-DM, the various deep learning models including Convolutional neural networks (CNN), CNN-OVA, RESNET50, and DENSENET121 are developed and trained to classify the skin cancer lesion images using the formatted data. The developed models are further enhanced by making fine adjustments to the models, to increase accuracy. The models that are developed and refined are evaluated for accuracy in the Evaluation phase, utilizing the metrics like F1 Score, recall, precision, and confusion matrix. This phase serves as the performance evaluator of the project. Deployment phase introduces the best performing model into the real-world by making the model accessible to the health care professionals, ensuring it can be used to make accurate and early detection of skin cancer class. All the findings are summarized in a detailed report.

Work Breakdown Structure

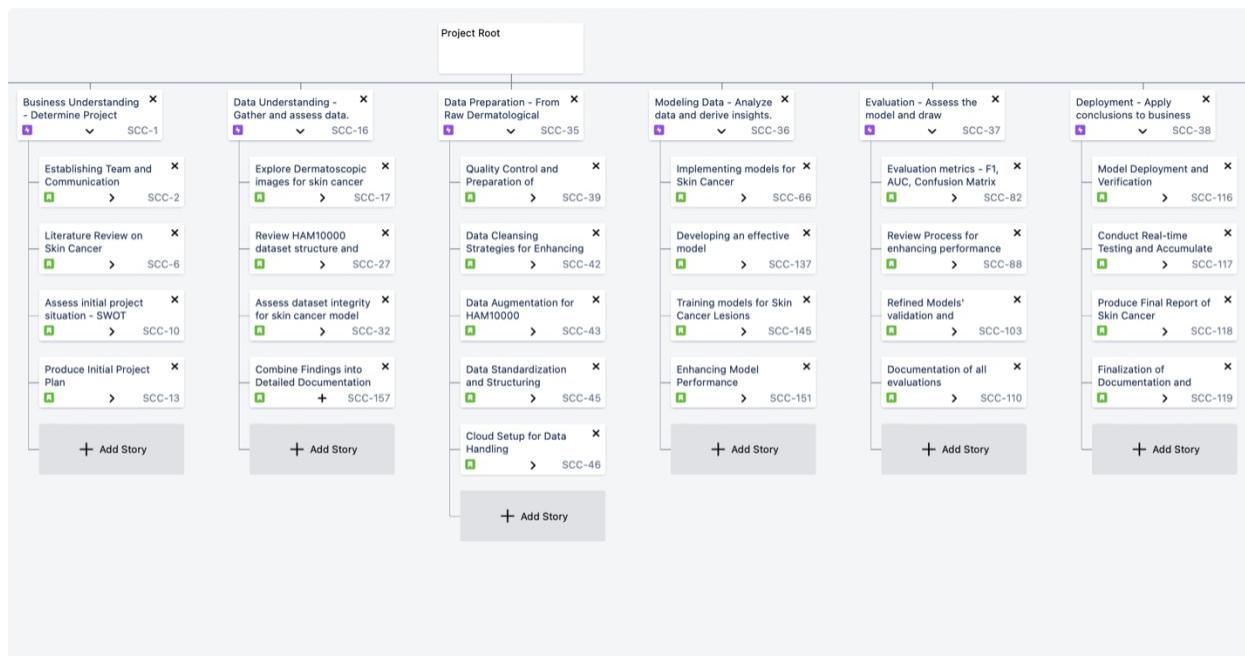
The Work Breakdown Structure (WBS) provides an organized plan for this research. This WBS outlines the major tasks and subtasks involved in a skin cancer classification project using deep learning, from project planning to post-deployment support and maintenance. Each task can be further decomposed into specific activities and assigned to team members accordingly for timely completion of the project. The objective of this research is to develop a model to classify the skin lesions accurately, utilizing the deep learning models.

In this project of skin cancer classification, WBS was meticulously designed to ensure timely completion and smooth progress. Beginning with one week dedicated to understanding the project's objectives and scope and spent another week for analyzing the HAM10000 dataset to

comprehend dermatoscopic image features. The subsequent two weeks will focus on data preparation, including refining images and extracting crucial features. Following this, allocated two weeks to construct and refine the models, incorporating CNN, DenseNet121, and ResNet50 architectures. The subsequent 10 days will involve evaluating and fine-tuning these models for optimal performance. Finally, in the last 12 days, models were deployed in a real-world environment to assess their effectiveness outside controlled conditions. The project will conclude with thorough documentation and a comprehensive final report.

Figure 7

Work Breakdown Structure



The structure of the WBS commences with Business Understanding Phase, which was decomposed into different deliverables and further into work packages. In this phase, the project's primary objective was defined, the project team was established, and communication channels were setup to ensure smooth collaboration. A comprehensive literature review on skin cancer classification was conducted, gathering insights from various sources such as

Google Scholar and Facebook Research. The initial project situation was assessed through a SWOT analysis, identifying strengths, weaknesses, opportunities, and threats. Subsequently, an initial project plan was produced, detailing timelines, resources, and milestones.

The next phase involves Data Understanding, where dermatoscopic images were explored for skin cancer. This included conducting a detailed analysis of images in the HAM10000 dataset to identify lesion locations and age features. The dataset structure and metadata were reviewed to understand various types of skin cancer lesion classes. Additionally, assessed the integrity of the dataset, ensuring completeness and quality for modeling purposes.

The raw data collected from the HAM10000 will undergo a series of preprocessing steps during the Data Preparation phase. This includes setting up a Google Cloud Platform environment to ensure efficient data storage and access. Additionally, data cleansing strategies will be applied to handle missing values and outliers, enhancing the overall quality of the dataset. To further improve skin cancer classification, data augmentation techniques using ImageDataGenerator will be utilized to enlarge and diversify the HAM10000 dataset. This will aid in developing models that are robust and consistent across various inputs. Lastly, the data will be standardized and structured, incorporating techniques like normalization to maintain uniformity across the dataset.

In the modeling phase of the project, began with setting up classification models such as CNN, CNN-OVA, DenseNet121, and ResNet50 to classify skin cancer lesions. Planned to work on the Google Colab Environment which allows us to utilize high level computational models with the help of GPUs (Kanani & Padole, 2019). This is followed by strategically developing these models through optimization of learning rates, weight initialization, and adjusting the number of layers to enhance their ability to classify skin cancer lesions accurately. Subsequently, these models, including CNN, CNN-OVA, DenseNet121, and ResNet50, will be trained on the

augmented HAM10000 dataset. The final step involves refining the models by adding regularization to the trained models to boost their performance and accuracy.

The evaluation phase includes assessing the performance of models trained on the HAM10000 dataset using metrics such as the F1 Score, precision, recall, and Confusion Matrix. This is followed by a review process to optimize the modeling process to improve performance, which involves understanding the configurations of hidden layers. After enhancing the models, their performance will be validated to ensure improved classification of skin cancer lesions. Finally, comprehensive documentation will be prepared to detail all evaluation processes and findings.

The final phase of WBS is the Deployment Phase, includes several important tasks beginning with the deploying of the best performed model into an operational setting to assess their performance in the realworld. The model will be tested on the dermatoscopic images provided by the health care professionals. Feedback will be collected from the end users to continuously monitor the model's performance. In the end, all the findings will be summarized in a detailed report.

Project Resource Requirements and Plan

For effective implementation of the skin cancer classification project, requires hardware infrastructure and sophisticated software tools. Key hardware includes a local machine with Intel Core i7 processor, NVIDIA RTX 30 series GPU for efficient deep learning model training, and 16GB RAM to handle large datasets. These specifications are crucial for training and fine-tuning deep learning models effectively, ensuring precise categorization of skin cancer lesions. The table provided outlines these essential hardware requirements.

The software requirements include the programming environment is Google Colab Pro, a cloud- based platform facilitating high-level computational tasks and collaborative programming.

Utilized Python 3.11 as main programming language, because of its ease of use and wide range of libraries. Crucial for data preprocessing, selected Sklearn 0.4.0 for its advanced machine learning algorithms, Pandas 1.3.3 for data manipulation and analysis, and Numpy 1.21.0 for numerical computations. Image processing, a vital component of project, is managed using TensorFlow 2.8.0, renowned for its powerful capabilities in developing and training machine learning models.

The key project management tools include JIRA, supporting Agile methodology, which was crucial for task management, sprint planning, and workflow review, including the creation of GANTT Charts and Work Breakdown Structures (WBS). Google Meet, a complimentary video conferencing platform, played a vital role in facilitating remote communication, especially for daily scrum meetings. Microsoft 365, provided for free with student IDs, was utilized for documentation needs.

Hardware Requirements

For the effective implementation of data preprocessing, modeling, and evaluating dermatoscopic images of Skin Cancer Classification project, needs a high-performance local machine with Intel i7 processor, model evaluation and other tasks, NVIDIA RTX 30 series GPU for training deep learning models which significantly helps in reduction of model training time, and 16 GB RAM to accommodate the huge datasets. The availability of these specifications is key to successfully train and finetune the deep learning models, to ensure accurately categorized skin cancer lesions. Table 3 outlines these hardware specifications required for this project.

Table 6*Hardware Specifications*

Hardware	Specifications	Cost
Intel i7 processor, NVIDIA RTX 30 series		
Local Machine	GPU, 16 GB RAM	USD 2300
Total		USD 2300

Software Requirements

The Skin Cancer Classification project requires the software libraries as mentioned in Table 4, where it clearly detailed the specific programming environment and libraries essential for the successful execution and development of project. The primary programming environment utilized was Google Colab Pro, which offers an advanced, cloud-based platform for executing high-level computational tasks. It also supports collaborative programming. Utilized Python version 3.11 as core programming language due to its ease of use, and versatility in handling various computational tasks providing a plethora of libraries.

For data preprocessing, which is a crucial step in project, a suite of libraries are chosen which are for their efficiency and functionality. This includes Sklearn version 0.4.0, known for its machine learning algorithms and tools, Pandas version 1.3.3, which is important for data manipulation and analysis, and Numpy version 1.21.0, essential for numerical computations. Image processing tasks, which form a significant component in project, utilize TensorFlow version 2.8.0. TensorFlow, known for its powerful and flexible tools, is ideal for developing and training machine learning models. Together, these tools and environments form the backbone of project's software infrastructure, ensuring efficient and effective development and execution of

project tasks.

Table 8

Software Requirements

Software	Libraries and their Versions	Cost
Programming environment	Google Colab Pro with subscription for two months	USD 19.98
Programming language Data Visualization Libraries	Python 3.11 Matplotlib 3.5.0, Seaborn 0.11.0	USD 0.00
Data Preprocessing Libraries	Sklearn 1.4.0, Pandas 1.3.3, Numpy 1.21.0	USD 0.00
Image processing libraries	Tensor Flow 2.8.0	USD 0.00
Total		USD 19.98

Tools and Licenses

For the successful and seamless execution of the project workflow and for the effective collaboration among the team members utilization of project management tools is inevitable. The project management tools used in this project include JIRA, Google Meet, and Microsoft 365. Table 5 summarizes all the essential tools and their costs.

JIRA is a project management tool that supports Agile methodology. It enabled to create, assign and track the progress of tasks and issues. It is also used to plan sprints, manage backlogs, and review workflows effectively. It enabled to create a GANTT Chart and WBS to

streamline the project workflow effectively. For this project, used a free version of JIRA.

Google Meet, a video conferencing platform was integral to this project to facilitate remote collaboration and communication within the team. This tool was essential in conducting DailyScrum meetings. For this project, a free version of Google Meet was used.

Microsoft 365 is a suite of productivity tools that was leveraged for documentation purposes in this project. This tool was made available for free with the student ID.

Table 9

Tools used

Tool	Configuration	Cost
Project Management	JIRA (free version)	USD 0.00
Project Collaboration	Google Meet (free version)	USD 0.00
Project Documentation	Microsoft 365 (free)	USD 0.00
Total		USD 0.00

Project Cost Estimation

The total budget allocated for the skin cancer classification project amounted to USD 2,319.98, as shown in Table 6. This amount includes both the hardware and software necessary to execute the project. The local machine with the requirements as Intel i7 processor for general tasks, an NVIDIA RTX 30 series GPU for efficient deep learning model training, and 16 GB RAM requires an amount of USD 2300.

The Google Cloud Platform, with USD 300 in the form of free credits was provided by Professor Edurado Chan, for each member of the team for 90 days. A budget of USD 19.98 was allocated to the usage of Google Colab Pro, it allows collaboratively to write and run the code for

the project.

Table 10

Cost Estimation

Type	Category	Estimated Cost
Hardware	Local Machine	USD 2300
Software	Programming Environment	USD 19.98
Software	Programming Language and Libraries	USD 0.00
Tools and Licenses	Project Management and Collaboration Tools	USD 0.00
Total		USD 2319.98

Project Schedule

Gantt Chart

A Gantt chart is a powerful project management tool that provides a visual representation of a project's timeline. Henry Gant developed this tool, which has proven to be important in project planning and tracking (Geraldi & Lechter, 2012). The chart organizes tasks or activities along the vertical axis, while the horizontal axis represents time intervals, providing a complete timeline from the beginning to the end of the task. Each task is shown as a bar, with length and orientation indicating the start, duration and completion of the project.

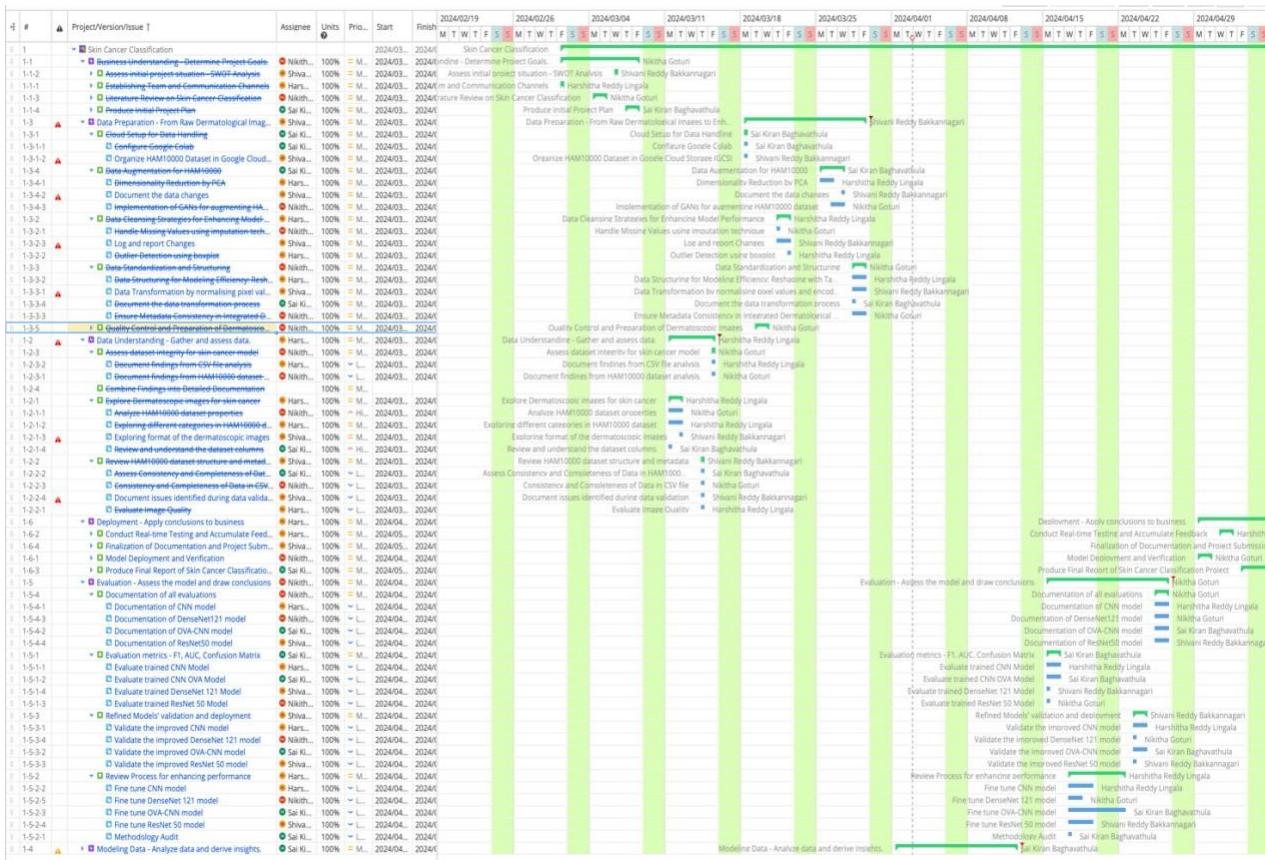
This graphically approached technique makes project managers and team members to easily understand company objectives, identify complicated milestones, and keep track of project progress in concise timelines to make quick and simple observations. Gantt chart is

important in making sure that all aspects of tasks are completed on time and effectively. Figure 7

illustrates the Gantt Chart of this study.

Figure 7

The Entire Gantt Chart



Every task in the project is represented by an individual horizontal bar spanning the length of its estimated duration, which is displayed against a carefully developed timetable. These charts excel at optimizing project plans by highlighting overlaps and interdependencies between tasks through smart placement on a shared timeline.

The distribution of particular duties to selected team members encourages individual accountability and promotes effective task management. Fundamentally, Gantt charts are an effective

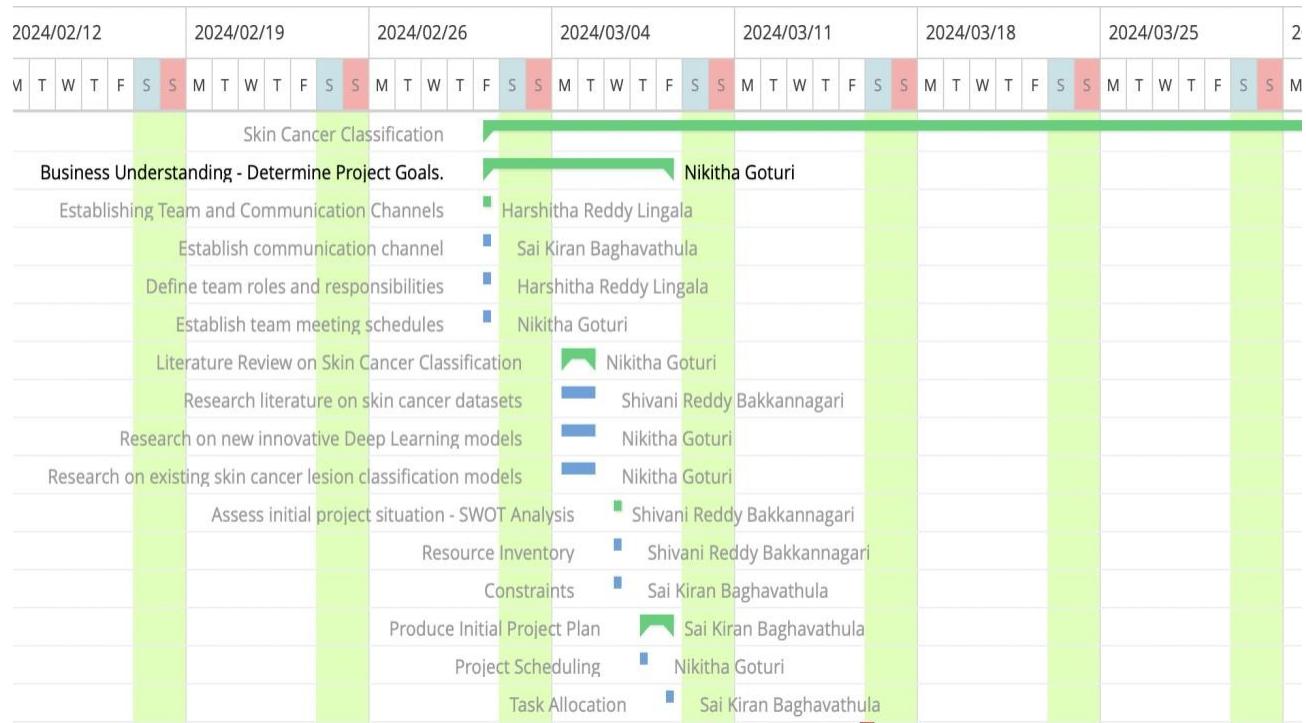
tool for thorough project planning and management because they give a clear, visual description of work divided across the project's lifespan.

Process development of the skin cancer classification project began with the business understanding phase, followed the established timeline, and continued through the implementation phase. Weekends were identified as non-working days consistent with formal organizational norms.

The breakdown of tasks was carefully reviewed and modified based on the complexity of each application, resulting in a strategy to meet the project objectives.

Business Understanding. During the Business Understanding phase, developed a base for skin cancer classification. During this phase project's goals, objectives, and stakeholder requirements were clearly mentioned. Then went through in-depth analysis to fully understand the scope of the project including its objectives and expected outcomes. The initial phase is very important in ensuring the subsequent efforts focused on accomplishing clearly defined goals that were compatible with the larger aims of skin cancer classification.

During this phase, Gantt chart was crafted to outline the project plan, highlighting key deliverables, defining the scope, and setting the timeline, as illustrated in Figure 8.

Figure 8*Representation of the Tasks in Business Understanding*

It also included assigning resources and identifying any limits or challenges to the timeline.

Overall, this phase provided a structure for the entire project by providing a clear knowledge of what must be done, as seen in Table 11.

Table 11*Business Understanding*

Task	Timeline	Team Member	Status of the Deliverables
Establishing Team and Communication Channels	03/01/2024 - 03/01/2024	Harshitha Reddy Lingala	Completed

Task	Timeline	Team Member	Status of the Deliverables
Literature Review on Skin Cancer Classification	03/04/2024 - 03/05/2024	Nikitha Goturi	Completed
Assess Initial Project Situation - SWOT Analysis	03/06/2024 - 03/06/2024	Shivani Reddy Bakkannagari	Completed
Produce Initial Project Plan	03/07/2024 - 03/08/2024	Sai Kiran Baghavathula	Completed

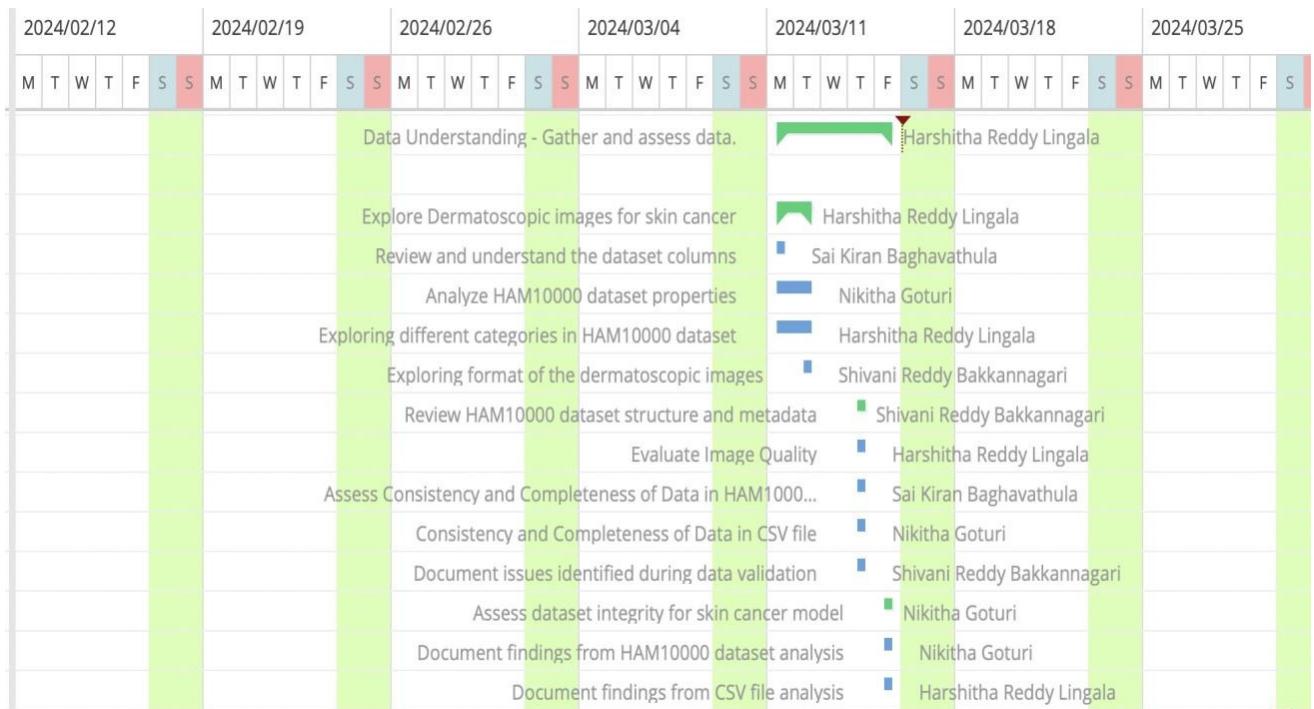
During this phase, focused on developing concise rules and regulations by conducting a thorough literature search to gain an in-depth knowledge of the project, by analyzing the current stage of the project using SWOT analysis and developing a well-structured initial plan. This phase is very important in focusing the efforts towards the project goals as well as for establishing foundation for focused and fruitful research strategy.

Data Understanding. The Data Understanding phase includes through study and evaluation of existing data to evaluate its characteristics, efficiency, and relevance to skin cancer classification project. This phase is very crucial for creating Gantt chart and for subsequent stages. It involves examining data sources, identifying its significance, limitations, biases and developing data collection processes. Initial evaluations of completeness and quality were also carried out, as shown in Table 12.

Table 12*Data Understanding*

Task	Timeline	Team Member	Status of the Deliverables
Explore Dermatoscopic Images for Skin Cancer	03/11/2024 - 03/12/2024	Harshitha Reddy Lingala	Completed
Assess Dataset Integrity for Skin Cancer Model	03/14/2024 - 03/14/2024	Shivani Reddy Bakkannagari	Completed
Combine Findings into Detailed Documentation	03/15/2024 - 03/15/2024	Nikitha Goturi	Completed

The Gantt chart extensively detailed the data collection phase including source identification and data acquisition, as well as the initial assessments. It was also scheduled for data cleansing, transformation, and preparation to ensure that the data was reliable and accurate for analysis. This phase is very important to evaluate the collected information and developing strategies for future project activities. The Figure 9 explains the detailed outline of the data exploration on HAM10000 Dataset

Figure 9*Representation of the Tasks in Data Understanding*

Data Preparation. This phase is focused on tasks such as finding and fixing the data quality concerns such as handling missing data, fixing the inconsistencies and to address the outliers. This also includes the techniques such as standardization format, data normalization, scaling, and transformation, all of which are very important for preparing data for analysis and modeling. Table 13 describes the detailed description of all these tasks.

Table 13*Data Preparation*

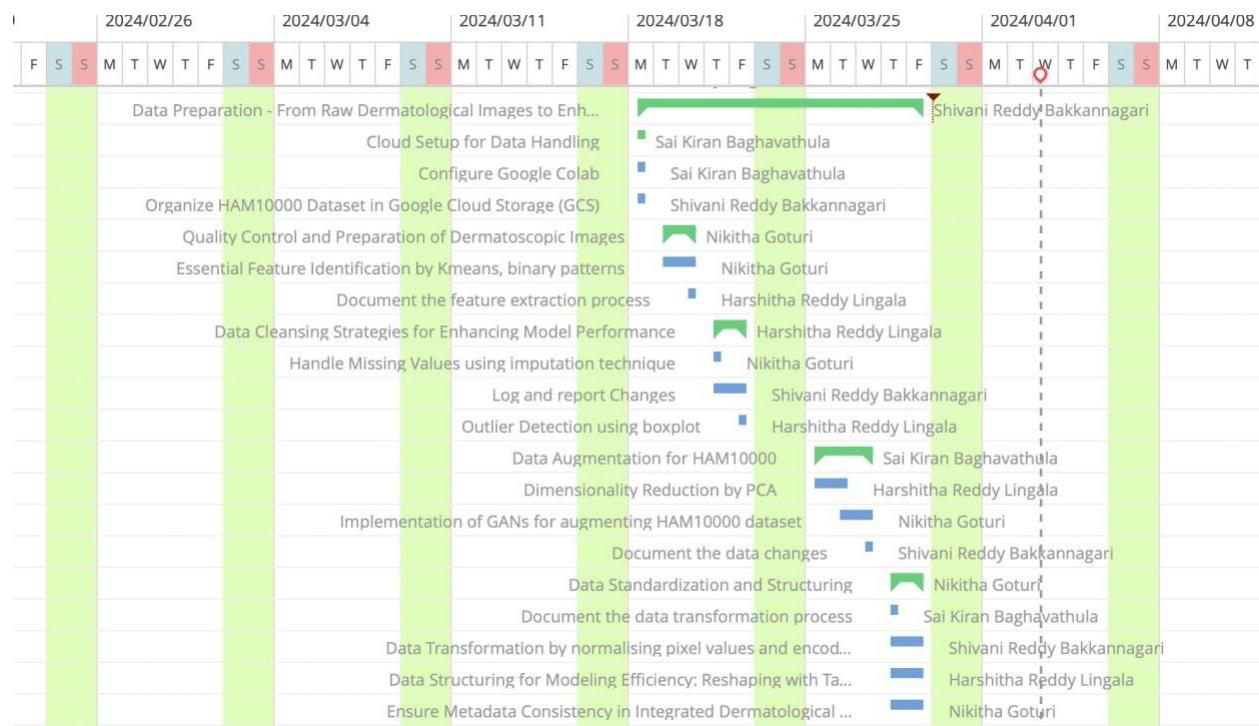
Task	Timeline	Team Member	Status
Review HAM10000 dataset structure and metadata	03/13/2024 03/13/2024	Sai Kiran Baghavathula	Completed
Cloud Setup for Data Handling	03/18/2024 - 03/18/2024	Sai Kiran Baghavathula	Completed
Quality Control and Preparation of Dermatoscopic Images	03/19/2024 - 03/20/2024	Nikitha Goturi	Completed
Data Cleansing Strategies for Enhancing Model Performance	03/21/2024 - 03/22/2024	Harshitha Reddy Lingala	Completed
Data Augmentation for HAM10000	03/25/2024 03/27/2024	Sai Kiran Baghavathula	Completed

This phase's Gantt chart identifies all critical activities such as data cleansing, data format standardization, and data transformation. This stage was set apart from rest by a great deal of work to fix data inconsistencies, feature selection, and data scale and standardization. Figure 10 describes the data preprocessing performed on the datasets, illustrating the normalization procedures and the

challenges of using data augmentation approaches.

Figure 10

Representation of the Tasks in Data Preparation

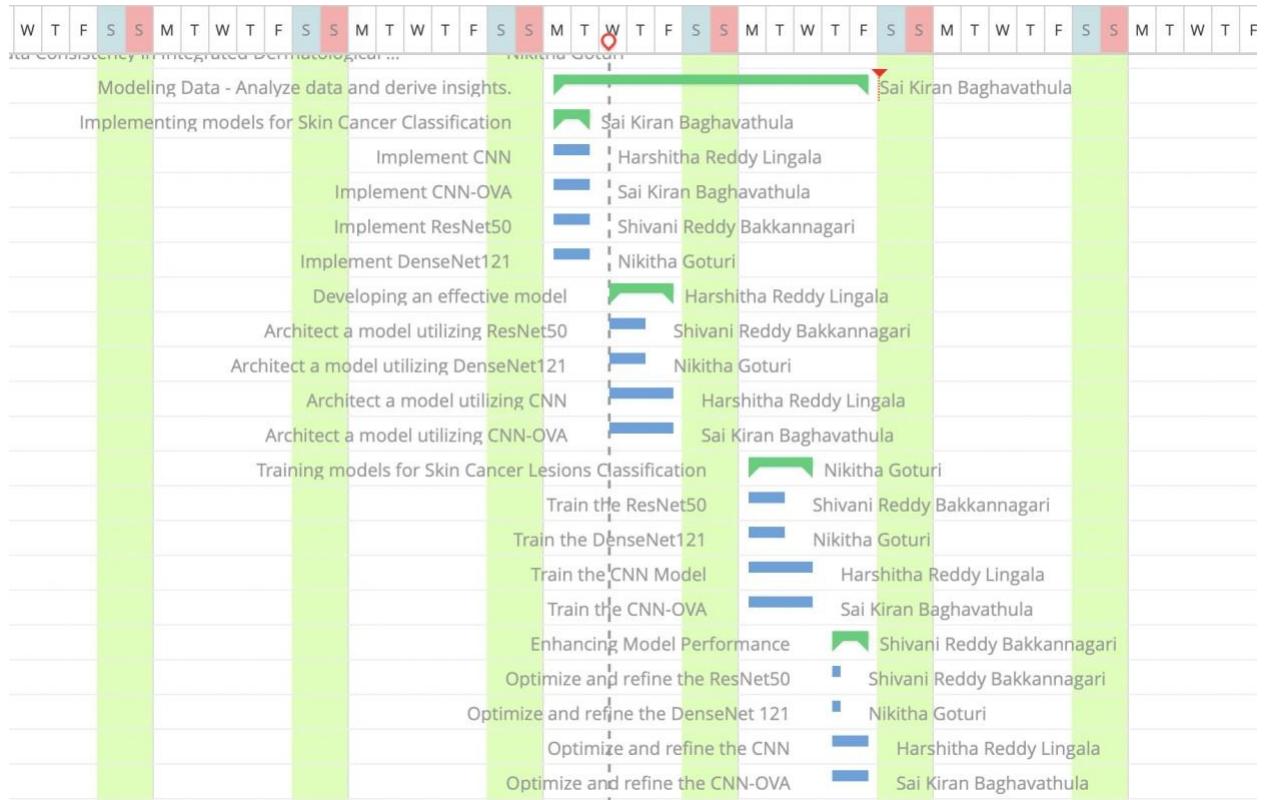


Modeling. During this phase, statistical techniques and complicated algorithms were used on the acquired information to create accurate models that fit the project objectives. This crucial phase is focused on identifying the best modeling technique for effectively analyzing the data and acquiring the relevant insights that could improve project's accuracy.

Table 14*Modeling*

Task	Timeline	Team Member	Status
Implementing Models for Skin Cancer Classification	04/01/2024 - 04/02/2024	Sai Kiran Baghavathula	Completed
Developing an Effective Model	04/03/2024 - 04/05/2024	Harshitha Reddy Lingala	Completed
Training Models for Skin Cancer Lesions Classification	04/08/2024 - 04/10/2024	Nikitha Goturi	Completed
Enhancing Model Performance	04/11/2024 - 04/12/2024	Shivani Reddy Bakkannagari	Completed

During the Modeling phase, the team developed various models including Convolutional Neural Networks (CNN), CNN with One-vs-All (CNN-OVA) approach, ResNet50, and DenseNet121. Each of these models were thoroughly trained and finely tuned to generate skin cancer lesion classification. The Gant chart clearly illustrated tasks such as identifying relevant approaches, dividing the data into training and testing datasets, building the models and engaging in iterative improvement and testing. It carefully planned different approaches for choosing appropriate modeling techniques, managing train and test data split, parameter adjustment and then testing. These steps were followed for building the most accurate and efficient models for skin cancer classification in accordance with the project's goal. Figure 11 displays the whole procedure.

Figure 11*Representation of the Tasks in Modeling*

Evaluation. The evaluation phase of the projects is very crucial since it is focused on evaluating the model's performance and adherence with the project objectives. Evaluation phase involves tasks including evaluating the models, implementing key performance metrics such as F1 Score, precision, recall, accuracy and Confusion Matrix. Assessing and improving efficiency via fine-tuning models based on evaluation results. Validating the upgraded models verified that they were ready for deployment and met the basic project requirements set during the Business Understanding phase.

The Gantt chart details tasks like performance evaluation, model comparisons, tuning for improved performance, and documentation which can be observed on Table 15.

Table 15*Evaluation*

Task	Timeline	Team Member	Status
Evaluation metrics -			
F1, Precision, Recall, Confusion Matrix	04/15/2024 04/16/2024	Sai Kiran Baghavathula	Completed
Review Process for Enhancing Performance			
	04/17/2024 - 04/22/2024	Harshitha Reddy Lingala	Completed
Refined Models' Validation and Deployment			
	04/23/2024 - 04/24/2024	Shivani Reddy Bakkannagari	Completed
Documentation of All Evaluations			
	04/25/2024 - 04/26/2024	Nikitha Goturi	Completed

The Gantt chart makes it simple to plan work like comparing models to the requirements established during the Business Understanding phase. The main tasks include completely testing and fine-tuning the models built in earlier phases, guaranteeing complete model validation, and documenting the results to verify the models' fit for deployment.

Figure 12

Representation of the Tasks in Evaluation phase



Deployment. The Deployment phase focused on translating successful models into practical applications. This involved preparing the models for deployment and integrating them into operational systems, as shown in Table 16.

Table 16*Deployment*

Task	Timeline	Team Member	Status
Model Deployment and Verification	04/29/2024 - 05/08/2024	Nikitha Goturi	Completed
Conduct Real-time Testing and Accumulate Feedback	04/29/2024 - 04/30/2024	Harshitha Reddy Lingala	Completed
Produce Final Report of Skin CancerClassification Project	05/01/2024 05/02/2024	Sai Kiran Baghavathula	Completed
Finalization of Documentation and Project Submission	05/03/2024 - 05/06/2024	Shivani Reddy Bakkannagari	Completed

This phase includes critical activities such as including model implementation, monitoring, documentation, and feedback loops for future enhancements. It includes establishing models for real-world use, providing recommendations and training, and continuous evaluation for efficacy.

Comprehensive documentation of model findings and preparation for the final project presentation were also required. Figure 12 shows a graphic explanation of these tasks.

Figure 12

Representation of the Tasks in Deployment



PERT

PERT (Program Evaluation Review Technique) charts to keep track of tasks and activities related to projects. PERT charts highlight task interdependencies and the critical route, in contrast to Gantt charts, which concentrate on project schedules (Evdokimov, 2018). The longest potential path is known as the critical path, and it is essential for estimating how long a project will take to complete.

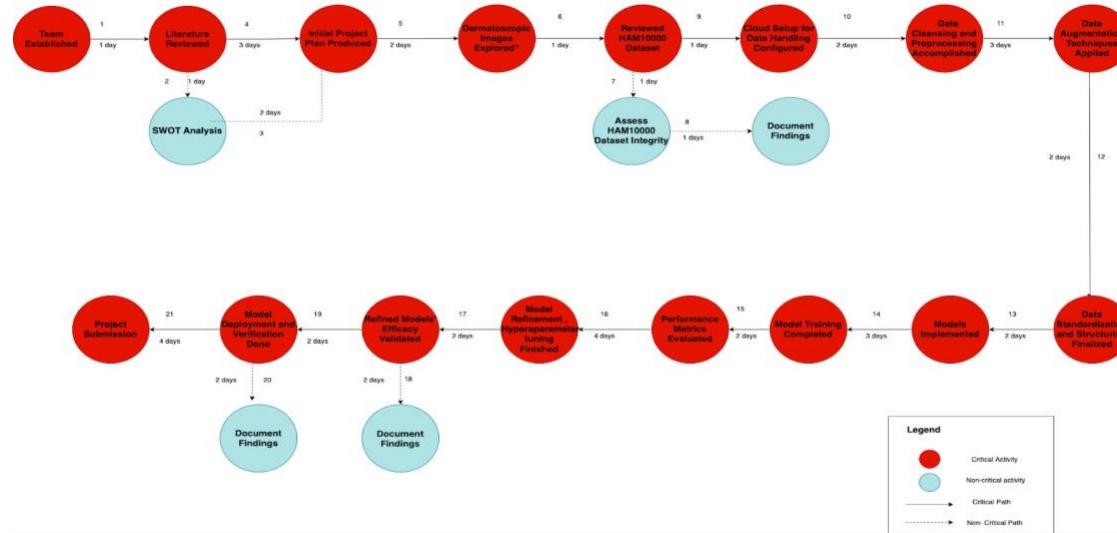
Task connections are displayed using arrows in PERT charts, and tasks and milestones are

represented by nodes. Firstly, a PERT chart containing tasks, dependencies, and time intervals is created to determine the critical path. After this, for better understanding and depth, this is aggregated at the milestone level.

Initially, the project's critical route was determined through the development of a Program Evaluation Review Technique (PERT) chart. The tasks required to complete the project were depicted graphically in this chart, as seen in Figure 13. Every work was specified, including the necessary conditions and the approximate time needed to finish it. Following the identification of each individual task and its corresponding length, the tasks were combined at the milestone level. This was carried out in order to improve clarity and deliver comprehensive insight into the project's development. With this method, the critical path, the order of tasks that determines the shortest time to finish the project could be shown. The activities of PERT are listed in Table 17.

Figure 13

PERT Chart (Milestone Level)



The PERT (Program Evaluation and Review Technique) chart, which focuses on 24 essential tasks, is an important tool for monitoring project progress in the skin cancer classification project. The chart is designed to avoid the common mistakes of micromanagement by demonstrating an extensive overview of the project's progress.

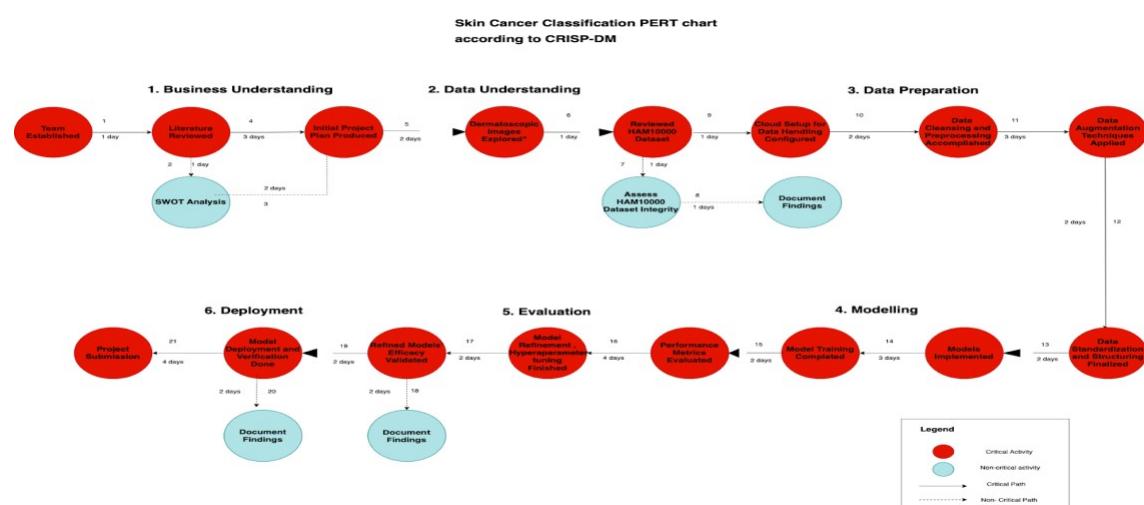
The critical path, which includes 19 of these activities (1, 4, 5, 6, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, and 21), is a precisely planned sequence that identifies the project's most important and time-sensitive sections.

The critical path begins with team building and progresses through activities such as literature analysis, dataset assessment, data preparation, and model implementation, ensuring that each phase is accomplished before moving on to the next. The skin cancer classification model's successful deployment and final presentation rely heavily on meticulous planning and structure.

Figure 14 depicts a pert chart developed with the CRISP-DM approach.

Figure 14

Pert chart according to CRISP-DM



As the project progresses, tasks such as "Domain Selection" and "Project Proposal for Skin Cancer Classification," help to establish the crucial route. These stages are critical in defining the project's objectives, ensuring they are explicit, quantifiable, attainable, relevant, and time bound. The path also covers crucial operations such as "Selecting Datasets," "Advanced Data Cleaning & Preprocessing," and "Feature Extraction," where the emphasis is on preparing and refining the data for in-depth analysis, as seen in Figure 14.

Table 17

Pert Activities with Description

Pert Activity	PERT Activity Description
1	Assigned team to do background research
2	Performed Opportunity and Risk Analysis
3	Project Proposal plan
4	Finalized the project proposal
5	Studied potential dermatoscopic image datasets
6	Finalized HAM10000 dataset
7	Verified data quality and consistency
8	Documented Findings
9	Configured cloud setup for data handling
10	Prepared data for modeling
11	Expanded scope of data to enhance training
12	Structure data for modelling

Pert Activity	PERT Activity Description
13	Crafted models for training
14	Fitted the models (CNN, CNN-OVA, DenseNet121, ResNet50)
15	Evaluated models against F1 score, precision, recall, accuracy metrics
16	Conducted iterative improvement based on performance analysis
17	Validated refined models
18	Documentation
19	Deployed models for real-time testing
20	Documentation
21	Project conclusion and Documentation

The advanced level of "Model Development: CNN, CNN-OVA, DenseNet121, ResNet50," where theoretical knowledge is translated into practical analytical tools for skin cancer classification, highlights the results of these efforts. This development pattern, as depicted on the PERT chart, emphasizes the significance of each phase in developing a reliable and precise skin cancer classification system.

The PERT chart for the Skin Cancer Classification project combines precise planning with the ability to respond to changes, resulting in effective management of this complicated activity. It's essential route, which represents the path from beginning to end, captures crucial actions during a 51-day period. Adhering to this approach is critical for the project to stay on track and meet its aim of developing an effective skin cancer classification system.

Data Engineering

Data Processing

Data Collection

The research utilizes publicly available open-source dataset, namely HAM10000 from Harvard Dataverse. This approach facilitates the reproducibility of the research's methodology and results, while also allowing for potential enhancements through future research efforts.

The HAM10000 dataset, also known as Human Against Machine, comprises approximately 10,015 dermatoscopic images in JPEG format. These images were collected over a span of 20 years from two locations: the Department of Dermatology at the Medical University of Vienna, Austria, and the skin cancer practice of Cliff Rosendahl in Queensland, Australia (Tschandl et al., 2018). The primary purpose of this dataset is for the development of deep learning models capable of identifying and classifying skin cancers from images.

Data Storage and Working Environment Setup

Given the significant importance of data in our project, it is crucial to prioritize its storage. Throughout the project we utilized Google Cloud Platform for efficient storage of 'HAM10000' dataset. Google Colab Pro is utilized for data preprocessing, model development, and evaluation. Deep learning networks in the project use sophisticated GPUs, Google Colab Pro emerges as the optimal solution. By integrating GitHub, a cost-free platform, with cloud resources, we established a robust methodology that can seamlessly integrate version control and collaborative development with efficient data management techniques.

Maintaining Data Quality and Integrity

Data is prepared for modeling following several data cleaning approaches, including handling the missing values for which we experimented with different imputation techniques and k-NN imputation is followed for replacing missing data after a comparative analysis. Presence of outliers,

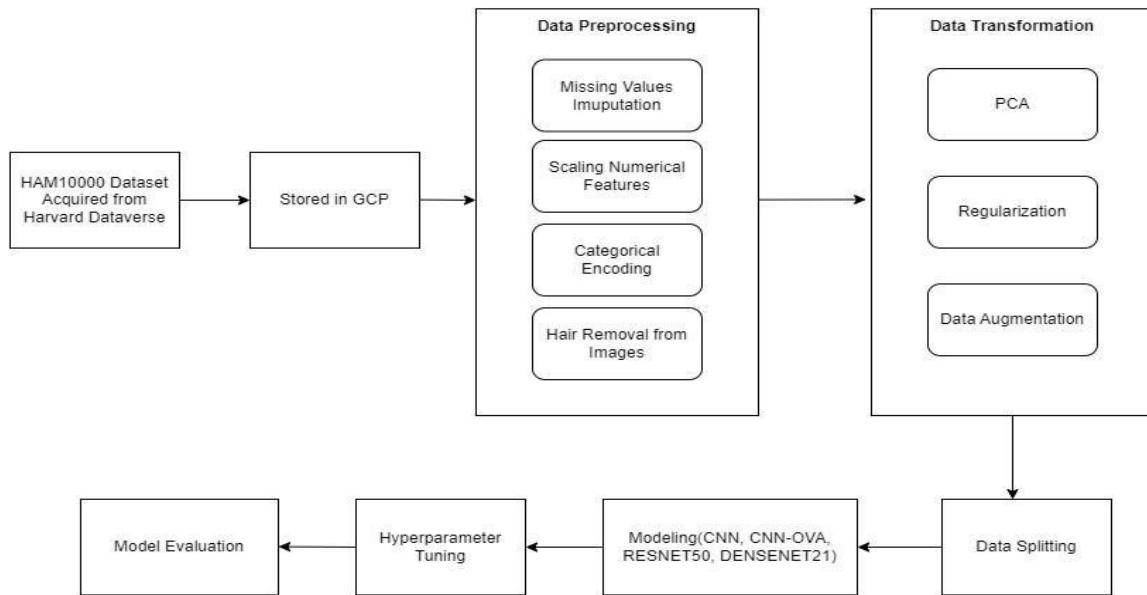
which can introduce bias during modeling, hence need to be handled during data preprocessing. The 'HAM10000' data is free outliers. However, we have proposed selective strategies that can be used to handle the outliers if present. Noise in the images is cleaned by removal of hair utilizing the Dull Razor Method to enhance the quality of the dermatoscopic images.

Scaling Numerical Features and encoding

Preprocessing techniques like normalization and standardization are crucial for enhancing machine learning model performance. We examined the data distribution to choose between normalization and scaling. It was identified that the data followed non-gaussian distribution, which led us to perform normalization on the cleaned dataset. Encoding categorical variables is crucial for machine learning, for processing of non-numeric data. In the HAM10000 dataset, 'sex' is binary and is label encoded ('male' as 0 and 'female' as 1) to maintain category distinction without adding dimensions. For non-ordinal categorical variables like 'localization' and 'dx type', one-hot encoding is performed.

Data Transformation and Data Splitting

Data augmentation is implemented to balance the dataset. Techniques like rotation augmentation, zoom augmentation and width shift augmentation. L2 regularization is employed to overcome the overfitting, by making the model to learn in a simpler and generalized way. For dimensionality reduction, PCA was implemented to reduce the dimensions. It reduced the dimensions of the dataset to 79. 60% of the dataset was used for training, 20% was used for testing and 20% was used for validation of the model performance.

Figure 15*Deep Learning Workflow of Skin Cancer Classification***Data Collection**

The research utilizes public, open-source HAM10000 dataset with a collection of 10,015 dermatoscopic images of pigmented skin lesions across seven diagnostic categories. The dataset includes images from diverse populations, using various modalities, and is confirmed through methods like histopathology and expert consensus. It's available for non-commercial use on platforms like Harvard Dataverse and Kaggle (Tschandl, P., 2018). The HAM10000 dataset contains 10015 files with seven lesion types. A more detailed description is illustrated in Figure 16.

Figure 16*HAM10000 Dataset*

metadata_df.head()						
Sample keys from image_paths: ['', 'ISIC_0024306', 'ISIC_0024307', 'ISIC_0024308', 'ISIC_0024309']						
Number of Null in image_path: 0						
lesion_id	image_id	dx	dx_type	age	sex	localization
0	HAM_0000118	ISIC_0027419	blk	histo	80.0	male
1	HAM_0000118	ISIC_0025030	blk	histo	80.0	male
2	HAM_0002730	ISIC_0026769	blk	histo	80.0	male
3	HAM_0002730	ISIC_0025661	blk	histo	80.0	male
4	HAM_0001466	ISIC_0031633	blk	histo	75.0	male
						scalp
						gs://ham10000dataset/HAM10000_images/ISIC_0027...
						gs://ham10000dataset/HAM10000_images/ISIC_0026...
						gs://ham10000dataset/HAM10000_images/ISIC_0025...
						gs://ham10000dataset/HAM10000_images/ISIC_0031...

In Figure 16, the DataFrame includes columns such as lesion_id, image_id, dx_type, age, sex, localization, and image_path. Each row corresponds to an individual lesion record. This structure is helpful for systematically processing and analyzing data. By relying on publicly available data like HAM10000 with detailed labels, the research aims to have an accessible and verifiable approach.

Table 18 below presents a comparative overview of key characteristics of HAM10000 dataset

Table 18

Summary of Key Features of Datasets

Dataset	HAM10000
Name	
Source of the dataset	https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/DBW86T
Files	HAM10000_images (folder contains images with unique image_id) and HAM10000_metadata.csv
Number of images	10015
Research Paper	The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions
Image Size	Each image varies in size, measured in kilobytes (KB)

It delineates the dataset's composition, encompassing image files stored in the "HAM10000_images" folder, alongside metadata in the "HAM10000_metadata.csv" file. With a total of 10,015 images varying in size (measured in KB), this collection offers a diverse array of dermatoscopic data crucial for comprehensive analyses in dermatology research.

Raw Data Samples

This section presents an in-depth examination of the raw data samples collected in the Study.

The data, consisting primarily of different images. Figure 17 presents detailed images from the HAM10000 dataset, specifically capturing the lesion of skin.

Figure 17

Dermatoscopic Variability in Skin Lesions



Figure 17 displays a series of dermatoscopic photographs, each showcasing different types of skin lesions. These images capture a variety of lesion characteristics such as

Color Variations. The lesions exhibit a range of colors from light brown to dark brown and red, indicative of different types of skin conditions.

Shape and Border Irregularity. Some lesions have regular, circular shapes, while others display irregular borders, which is a significant factor in skin cancer diagnostics.

Texture and Pattern. The texture varies among the lesions; some have a smooth appearance, while others show rough or scaly textures. The patterns within the lesions, such as network-like structures or homogeneous areas, are critical for identifying specific types of melanoma or other skin disorders.

Hair Presence. In some images, hair follicles are visible, which can influence the analysis of the skin's surface texture and the lesion's characteristics.

The data collection approach for the HAM10000 dataset encompasses a variety of methodologies providing a comprehensive overview of the data acquisition process. It is illustrated in Figure 18.

Figure 18

Data Collection Plan

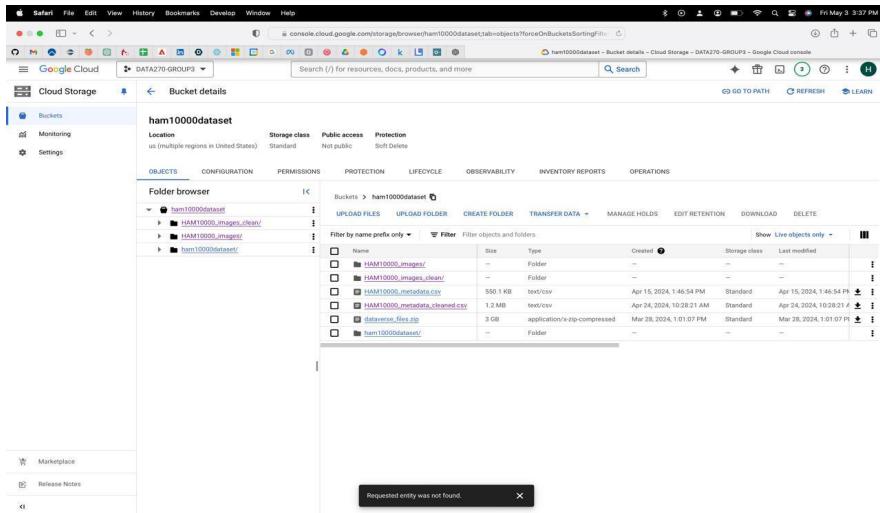
DATA COLLECTION PLAN							
Project Number:	Group 3		Date:	4/18/24			
Project Title:	Skin Cancer Classification Using Deep Learning			Project Leader:			
Description of the data collection plan							
We obtained the 'HAM10000' dataset from Harvard Dataverse for training, comprising 10,015 dermatoscopic images in JPEG format. These images were gathered from diverse populations over a span of 20 years from two different sites - the Department of Dermatology at the Medical University of Vienna, Austria, and the skin cancer practice of Cliff Rosenthal in Queensland, Australia, and were collected from the era before the availability of digital cameras.							
What will be done with the data once it has been collected							
1. Explore the HAM10000 Data		5. Use the formatted data for Model Training(CNN, RESNET50, DENSENET121, CNN-OVA) 6. Evaluate the Models against metrics F1 Score, Accurac, AUC, and Confusion Matrix 7. Fine tune the models 8. Detailed documentation of the findings					
Key Variables - A summary of the chosen input variables(X's) and output variables(Y's)							
	1	2	3	4	5	6	7
Variable Title	lesion_id	image_id	dx_type	age	sex	localization	dx
I/p or o/p variable	X	X	X	X	X	X	Y
Unit of Measurement	N/A	N/A	N/A	years	N/A	N/A	N/A
Data Type	Categorical	Categorical	Categorical	Continuous	Categorical	Categorical	Categorical
Collection Method	Clinical Diagnosis	Digital Imaging	Clinical Diagnosis	Registration Data	Registration Data	Registration Data	Clinical Diagnosis
Gauge Instrument	Dermatologist	Dermatoscope	Dermatologist	Administrative Records	Administrative Records	Administrative Records	Dermatologist
Location	Clinics USA	Clinics USA	Clinics USA	Clinics USA	Clinics USA	Clinics USA	Clinics USA
Gauge Calibrated	N/A	Yes	N/A	N/A	N/A	N/A	N/A
Measurement System Checked	N/A	Yes	Yes	Yes	Yes	Yes	Yes
Precision (R & R) Adequate	N/A	Yes	N/A	N/A	N/A	N/A	N/A
Accuracy Adequate	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Historical data exist?	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Source of Historical Data	Clinics Records	Dermatology Clinics	Clinics Records	Hospital Records	Hospital Records	Hospital Records	Clinics Records
Historical data representative/reliable	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Mean	N/A	N/A	N/A	51.86	N/A	N/A	N/A
Upper Specification Limit	N/A	N/A	N/A	85	N/A	N/A	N/A
Lower Specification Limit	N/A	N/A	N/A	0	N/A	N/A	N/A
Standard Deviation	N/A	N/A	N/A	16.97	N/A	N/A	N/A
Target	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Minimum sample size	10015	10015	10015	10015	10015	10015	10015
sampling frequency	Once	Once	Once	Once	Once	Once	Once
sub-grouping needed	No	No	No	No	No	No	No
sub-group size	N/A	N/A	N/A	N/A	N/A	N/A	N/A
stratification needed	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Data collector	Harshitha Reddy	Nikitha Goturi	Sai Kiran Bhagavathula	Shivani Reddy Bakannagar	Harshitha Reddy	Nikitha Goturi	Sai Kiran Bhagavathula
Operational definition exist?	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Data Collector trained?	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Resources available for data collector?	Yes	Yes	Yes	Yes	Yes	Yes	Yes
When?	Start Date	3/11/24	3/11/24	3/11/24	3/11/24	3/11/24	3/11/24
	Due Date	3/15/24	3/15/24	3/15/24	3/15/24	3/15/24	3/15/24
	Duration	5	5	5	5	5	5

This template outlines the Data Collection Plan for the Skin Cancer Classification project using Deep Learning techniques. The project leverages the HAM10000 dataset, which consists of dermatoscopic images aimed at classifying skin cancer types through advanced machine learning models.

The data collection plan is meticulously designed to prepare the dataset for effective model training and evaluation. The following steps are included in the data handling and processing stages: Dataset Exploration, Missing Data Handling, Data Standardization, Data Augmentation, Model Training, Model Evaluation, Model Fine-Tuning, Documentation.

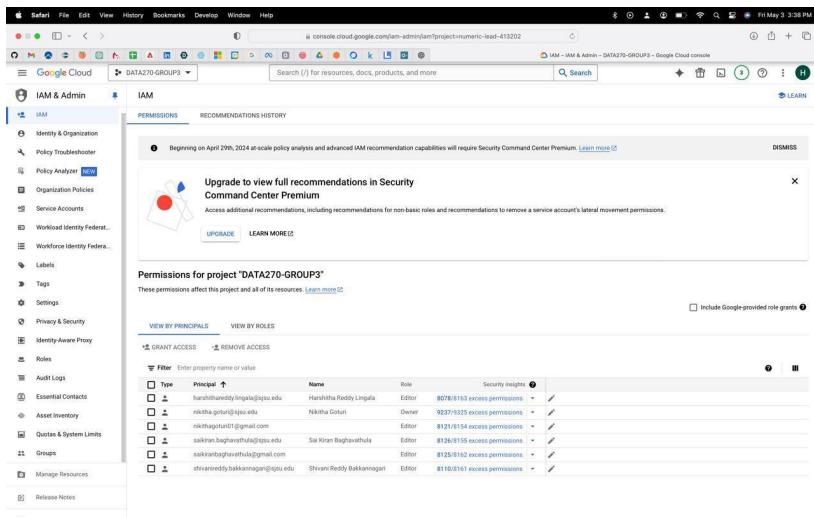
Data collection procedures for data standardization, augmentation, and the evaluation metrics used, such as F1 Score and AUC, to ensure the reliability and accuracy of the data used in developing skin cancer classification models. The data set's integrity seems to be underscored by rigorous checks and validation steps mentioned, such as gauge calibration of the dermatoscope and consistent measurement system checks, which ensure the precision and accuracy of the data collection process. This comprehensive approach confirms that the dataset meets the necessary standards for use in developing reliable models, like the detailed and systematic process you described.

After the data collection, stored the acquired datasets in a secure Google Cloud bucket as illustrated in Figure 19.

Figure 19*Google Cloud Bucket*

For their scalability, security, and simplicity of use. We chose Google Cloud storage for managing massive datasets due to its scalability, security, simplicity of access, and rapid data retrieval throughout processing stages.

Only authorized team members, as shown in Figure 20, have access to the cloud storage, which follows stringent regulations to ensure data integrity and security.

Figure 20*Access Privileges*

Controlled access enables distant and collaborative work, allowing team members to retrieve and analyze data simultaneously.

Data Pre-processing

The 'HAM10000_metadata.csv' file for the HAM10000 image dataset contains essential information like lesion ID, image ID, diagnosis, diagnosis type, age, sex, and localization. This metadata plays a crucial role in skin cancer classification using deep learning. It helps integrate diagnostic and demographic information with respective image data, facilitating supervised learning for predicting skin cancer class. By integrating this metadata with image data, researchers can develop more accurate deep learning models for skin cancer classification. It is important to ensure that the metadata is free from missing values and inconsistency, such that it can improve the integrity of the dataset, prevent bias in model predictions, and enable more reliable analysis.

Handling Missing Data

The 'HAM10000_metadata.csv' is studied thoroughly, to filter for missing values. Kang (2013) describes that missing data poses several challenges such as diminishing statistical power, which is the likelihood of correctly rejecting a false null hypothesis. Additionally, it introduces bias in parameter estimation and compromises the representativeness of samples. Furthermore, it complicates study analysis, jeopardizing the validity of trials and potentially leading to erroneous conclusions. This leads us to find a suitable way to handle the missing data.

Figure 21

Count of Missing values in the metadata

```
→ Count of Missing Values in the HAM10000 dataset
  lesion_id      0
  image_id       0
  dx            0
  dx_type       0
  age           57
  sex            0
  localization   0
  dtype: int64
```

The 'age' column has entries '0', which could have many explanations. Firstly, '0' might signify data entry errors or missing information. Secondly, it could indicate cases where the patient's age is unknown. Thirdly, '0' might represent infants or newborns, although skin cancer is exceedingly rare in this age group. To ensure unbiased classification, we propose treating 'age' values of '0' as indicating missing data. Following this replacement, the Python library is utilized to identify the overall count of missing values, as illustrated in Figure 22.

Figure 22

Count of Missing Values after modifying 'age' column

```
Count of the Missing values in the HAM10000 dataset after 'modifying the 'age' column
lesion_id      0
image_id       0
dx            0
dx_type       0
age           96
sex            0
localization   0
dtype: int64
```

There are number of ways that can be implemented to handle missing values in a dataset such as 'HAM10000', some of them are mentioned below:

Deletion of Entries with Missing Values. Deletion is one of the methods which disregards the data. This leads to information loss and introduction of bias, especially when data is not missing completely at random (Emmanuel et al., 2021).

Mean, Median, and Mode Imputation. This approach of handling missing values involves replacing the nulls with mean or median in case of continuous variables, and mode for categorical variables. However, following this method of simple imputation can introduce bias into data, causing inaccurate predictions.

KNN Imputation. KNN imputation finds the k closest complete instances to a missing data point within a dataset. If the target feature is categorical, it fills in the missing data with the most common

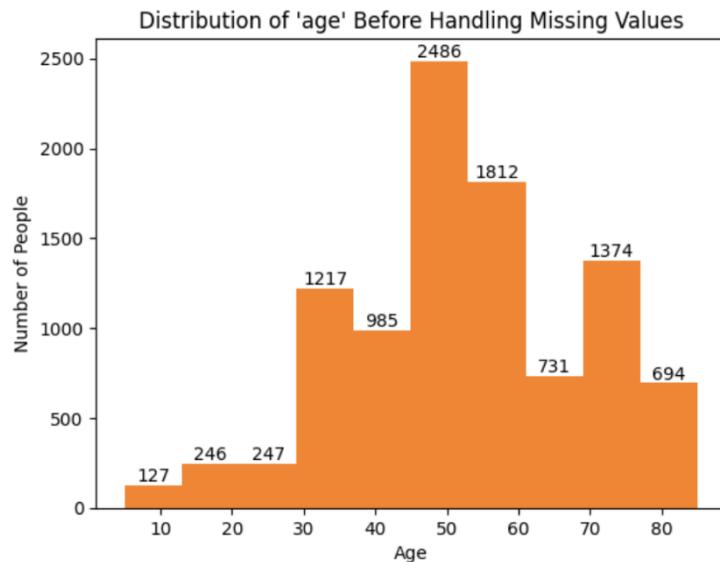
value among those neighbors. For numerical features, it takes the average of the neighboring values to fill in the missing data (Zhang, 2012).

Decision Tree. In the decision tree method, missing values for each variable are filled in by constructing decision trees based on available data. Predictions for missing values are then made using the corresponding tree's outcomes. Random Forest, a variant of this method, combines multiple decision trees to aggregate predictions (Emmanuel et al., 2021).

In our research of classifying skin cancer lesion images, we investigated the most effective imputation techniques, specifically KNN and Random Forest imputation, to predict the missing values effectively. Figure 23 illustrates the distribution of data after filling in the missing values.

Figure 23

Plot of Distribution of Age in dataset with Missing Values

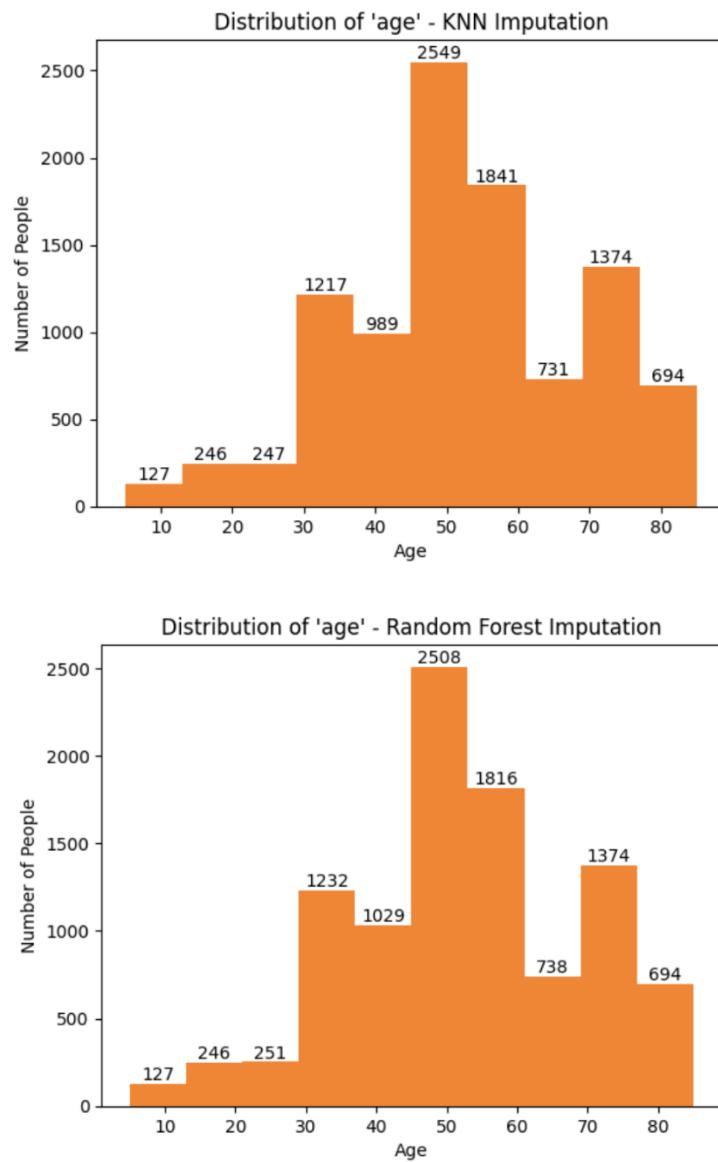


To understand the effectiveness of the two different missing value imputation techniques, we will employ both KNN (k-nearest neighbors) and random forest imputation on the "age" variable within our dataset separately. We will then compare the resulting age distributions post-imputation. Figure shows comparison of both the imputation techniques. This comparative analysis will enable us to

determine the imputation method that best preserves the original distribution of age data, ultimately guiding our selection of the optimal imputation strategy.

Figure 24

Plot of Distribution of Age in dataset comparing KNN and Random Forest Imputations



Following the comparison, if there is minimal difference between the imputed age distributions and the original age distribution for both techniques, the choice will depend on interpretability. In this case, we opt for KNN, as it's a simpler algorithm to understand for missing value imputation.

Handling Outliers in the Data

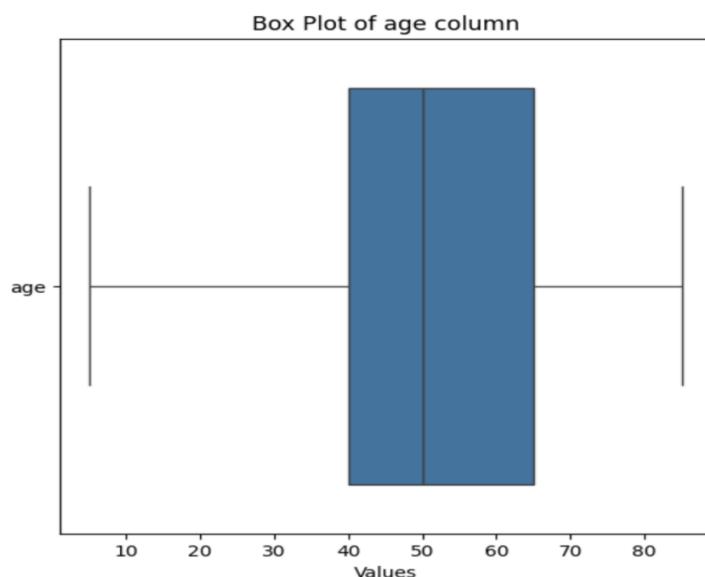
Outliers represent extreme values or errors in the data, which can significantly distort machine learning models, leading to incorrect classification. Identifying and appropriately handling outliers in medical datasets is essential to ensure accurate and reliable outcomes.

Box plots visually represent the distribution of data by displaying the interquartile range (IQR) as a box, with the median marked inside. The whiskers extend to 1.5 times the IQR from the edges of the box, and individual points beyond the whiskers are considered potential outliers, visually distinguishing them from the rest of the data. This visualization enables quick identification of outliers, helping in maintaining the data quality.

In the HAM10000 dataset, outliers are not present, as illustrated in Figure 25. However, it is typical for medical datasets to include outliers, like other types of data. Handling outliers can be approached similarly to how missing values are addressed. For continuous variables, mean imputation can be employed, while mode imputation is suitable for categorical variables. For more accurate imputation, predictive models such as KNN or Random Forest Regressor can be utilized.

Figure 25

Box plot of column ‘age’ to detect outliers



Best Practices to Handle the Outliers. Dash et al. (2023) proposes two approaches to handling outliers. Firstly, the strategy of addressing error outliers, where the focus is on either rectifying or eliminating erroneous data points. This involves careful consideration: if there's valuable information within these outliers that shouldn't be lost, efforts should be made to correct them rather than deleting them. However, if the errors are significant, deletion might be the reliable approach. Secondly, there's the handling of non-error outliers, which presents its own set of challenges. Here, three main methods come into play: keeping, deleting, or encoding. Keeping outliers requires caution, as they can skew the results of analyses or tasks at hand. Deleting them is a straightforward option but may lead to the loss of useful information. Recoding, on the other hand, offers a compromise by using techniques like winsorizing to mitigate the impact of outliers while retaining most of the data. Each approach has its trade-offs, and the choice depends on the specific context and objectives of the analysis.

Encoding

Encoding categorical variables is essential for machine learning because it enables algorithms to process non-numeric data effectively. By converting categorical labels into numerical representations, we ensure compatibility with machine learning algorithms while avoiding biases and preserving the integrity of the original data.

The HAM10000 dataset needs columns, 'sex', 'dx type', and 'localization' to be encoded. For the binary 'sex' column, label encoding offers a straightforward solution. With only two distinct categories, 'male' and 'female', label encoding assigns a unique numerical label to each category ('male' as 0 and 'female' as 1), preserving the distinction between them without introducing additional dimensions. This method effectively transforms the categorical data into a format that machine learning algorithms can readily interpret.

On the other hand, for categorical variables with multiple non-ordinal categories, such as the 'localization' and 'dx type' columns, one-hot encoding is preferable. With one-hot encoding, each

category is represented by a binary column, where a value of 1 indicates the presence of the category and 0 indicates its absence. This ensures that there are no unintended ordinal relationships between categories and maintains the independence of each category within the dataset. While one-hot encoding does increase the dimensionality of the dataset, it is a crucial step for accurately representing categorical variables and avoiding biases in machine learning models.

By employing label encoding for the 'sex' column and one-hot encoding for the 'localization' and 'dx type' columns, we effectively prepare the data for machine learning analysis. These encoding techniques ensure that categorical variables are appropriately represented, enabling machine learning algorithms to extract meaningful insights and make accurate predictions based on the data.

Figure 26

HAM10000 Dataset (5 rows) before Encoding

	lesion_id	image_id	dx	dx_type	age	sex	localization
	lesion_id	image_id	dx	dx_type	age	sex	localization
0	HAM_0000118	ISIC_0027419	bkl	histo	80.0	male	scalp
1	HAM_0000118	ISIC_0025030	bkl	histo	80.0	male	scalp
2	HAM_0002730	ISIC_0026769	bkl	histo	80.0	male	scalp
3	HAM_0002730	ISIC_0025661	bkl	histo	80.0	male	scalp
4	HAM_0001466	ISIC_0031633	bkl	histo	75.0	male	ear

Figure 27

HAM10000 Dataset (5 rows) after Encoding

Figure 28

HAM10000 Dataset (5 rows) after Encoding

	age	lesion_id	image_id	dx	dx_type	sex	0	1	2	3	...	5	6	7	8	9	10	11	12	13	14
0	80.0	118	3113	2	3	1	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
1	80.0	118	724	2	3	1	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
2	80.0	2710	2463	2	3	1	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
3	80.0	2710	1355	2	3	1	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
4	75.0	1460	7327	2	3	1	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Scaling Numerical Features through Normalization

Preprocessing techniques transform data into a consistent format and improve the performance of machine learning models, and the techniques include normalization and standardization. These methods adjust the scale and distribution of features to mitigate issues caused by varying magnitudes and distributions in the data. Normalization scales the data to a specified range, typically between 0 and 1, ensuring that each feature contributes equally to the analysis. Standardization transforms the data to have a mean of 0 and a standard deviation of 1, preserving the relative differences between data points.

The choice between normalization or scaling depends on the distribution of the data. If the data does not have outliers and follows a Gaussian distribution, StandardScaler might be more appropriate. On the other hand, if the data has to be within a certain range, we can normalize the data using MinMaxScaler().

To understand the normality of the data shown in Figure 15, we performed both the Shapiro-Wilk test and the Kolmogorov-Smirnov test for normality. The p-values from both tests are collected for each column. Since all p-values are not greater than the significance level alpha (usually set to 0.05), indicating that the data does not follow a Gaussian distribution, normalization is applied using MinMaxScaler which is evident from Figure 29.

Figure 29

HAM1000 dataset after normalization

	age	lesion_id	image_id	dx	dx_type	sex	0	1	2	\		
0	0.9375	0.015799	0.310865	0.333333	1.0	0.5	0.0	0.0	0.0			
1	0.9375	0.015799	0.072299	0.333333	1.0	0.5	0.0	0.0	0.0			
2	0.9375	0.362833	0.245956	0.333333	1.0	0.5	0.0	0.0	0.0			
3	0.9375	0.362833	0.135311	0.333333	1.0	0.5	0.0	0.0	0.0			
4	0.8750	0.195475	0.731676	0.333333	1.0	0.5	0.0	0.0	0.0			
...		
10010	0.4375	0.380774	0.876573	0.000000	1.0	0.5	1.0	0.0	0.0			
10011	0.4375	0.380774	0.923108	0.000000	1.0	0.5	1.0	0.0	0.0			
10012	0.4375	0.380774	0.921710	0.000000	1.0	0.5	1.0	0.0	0.0			
10013	0.9375	0.031999	0.853605	0.000000	1.0	0.5	0.0	0.0	0.0			
10014	0.8125	0.466863	0.794088	0.666667	1.0	0.0	0.0	0.0	1.0			
	3	...	5	6	7	8	9	10	11	12	13	14
0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
1	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
2	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
3	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
4	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...
10010	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10011	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10012	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10013	0.0	...	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10014	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

[10015 rows x 21 columns]

Handling Data Inconsistency

Data inconsistency occurs when there are conflicts either within a single dataset or across multiple datasets when merged from different sources (Agomor & Appiah, 2023). These inconsistencies can arise as differences in metadata, entity identification, or conflicting data values due to different representations or scales.

Data inconsistency can occur due to schema integration issues, such as merging metadata from different sources, or challenges in entity identification, where entities may be represented differently across datasets to be merged. Conflicts also arise from variations in data values, such as variations in measurement units (e.g., metric vs. imperial units). The ‘HAM10000’ dataset is free of data inconsistencies. However, we propose strategies that can be implemented to handle the data inconsistency.

To address data inconsistency, methods like schema integration are employed to reconcile metadata across sources. Entity identification techniques help unify entities with different

representations. Resolving conflicts in data values involves techniques like correlation and covariance analysis, which identify redundant attributes and assess their relationships to mitigate redundancy and ensure data coherence.

Handling Noisy Data

The HAM10000 dataset contains unwanted noisy elements such as hairs and shadows, which can negatively impact feature extraction and ultimately affect the performance of the classification model. To mitigate this issue, a hair removal algorithm, known as the Dull Razor Method, was applied to preprocess the images.

Hair removal algorithm, Dull Razor Method is a series of morphological operations used to detect and remove human hairs from images. The algorithm consists of the following steps:

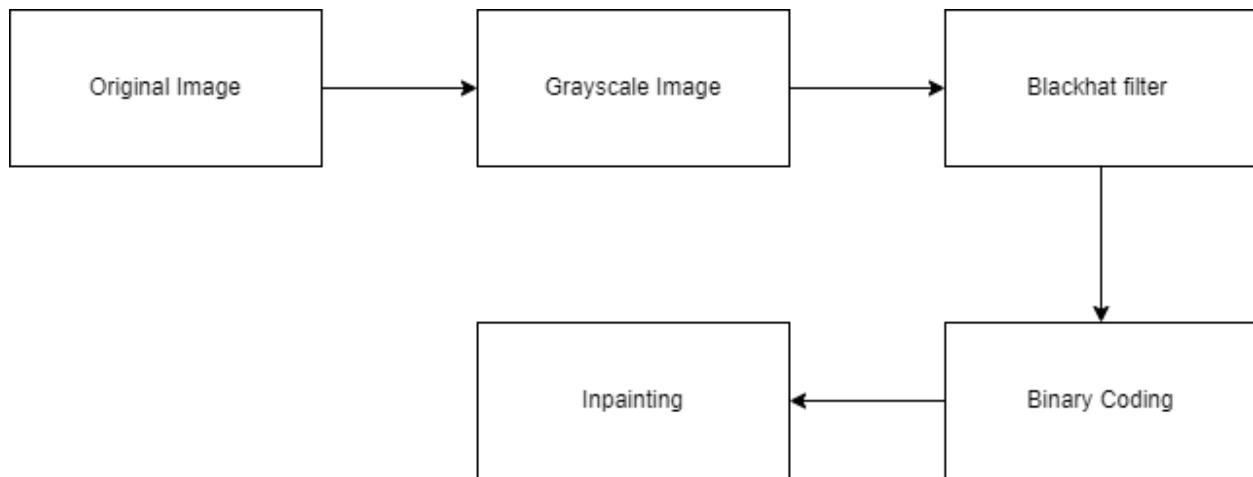
Image Loading and Gray scaling. The original image is loaded and converted to grayscale to facilitate hair identification.

Morphological Operations. A Black Hat morphological operation is applied to the grayscale image using a kernel, which highlights dark colors on light backgrounds, effectively highlighting the hairs.

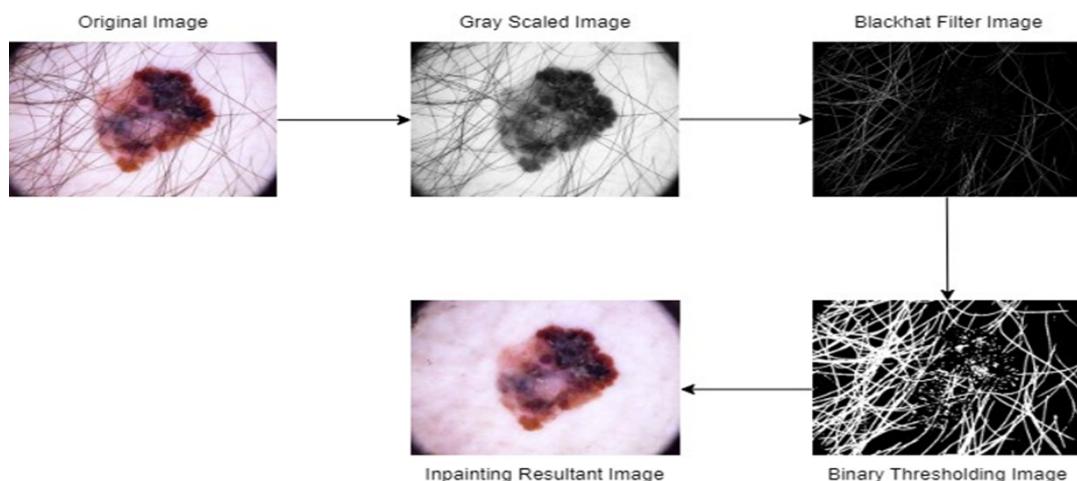
Binary Thresholding. The resulting image is then subjected to binary thresholding, where pixels with values greater than 10 are set to 255 (white), and all other pixels are set to 0 (black), effectively isolating hair regions. This creates a mask of hair regions that enable hair removal.

Inpainting. The hair region is then filled with surrounding pixels, effectively replacing the hair features with the surrounding skin texture. By applying the Dull Razor Method, unwanted hair features are removed from the images, enhancing the quality of the dataset, and improving the accuracy of the skin cancer classification model.

By applying the Dull Razor Method, unwanted hair features are removed from the images, enhancing the quality of the dataset, and improving the accuracy of the skin cancer classification model. The flow of this method is shown in the Figure 30.

Figure 30*Dull Razor Algorithm Flow*

A visual representation of the images obtained at different stages throughout the application of the Dull Razor Method is provided in the Figure 16. It illustrates how the unwanted hair features are progressively removed from the images, showcasing the transformation and refinement process as each step is carried out. This visual aid allows for a clear understanding of the method's effectiveness in enhancing the quality of the dataset for improved accuracy in skin cancer classification.

Figure 31*Resulting Images from Dull Razor Hair Removal Algorithm*

Data Transformation

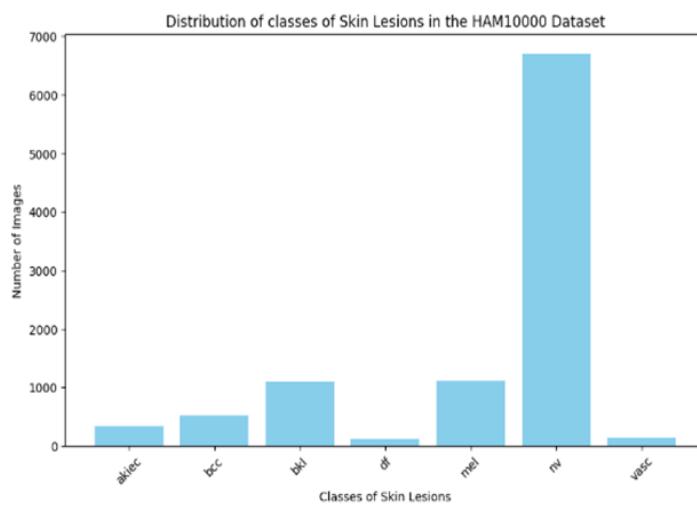
Data Augmentation

The HAM10000 dataset (Tschandl, 2018) consists of dermatoscopic images categorized into seven classes of skin lesions, with sample distribution as follows: Actinic keratoses (akiec) with 327, Basal cell carcinoma (bcc) with 514, Benign keratosis-like lesions (bkl) with 1099, Dermatofibroma (df) with 115, Melanoma (mel) with 1113, Melanocytic nevi (nv) with 6705, and Vascular lesions (vasc) with 142. The classes like Dermatofibroma, Basal cell carcinoma, Actinic keratoses, Vascular lesions, Benign keratosis-like lesions and Melanoma have significantly fewer samples than Melanocytic nevi class. This imbalance can have a negative impact on the performance of machine learning models, especially in classes with limited representation.

Figure 32 shows the class distribution before data augmentation, it is evident that the numbers are distinctly out of balance.

Figure 32

Class Distribution Before Augmentation



To mitigate this issue, data augmentation methods were utilized to expand the number of samples in underrepresented classes to balance the dataset. This augmentation not only rectified the imbalance issue but also improved the overall robustness and reliability of our classification model.

Augmentation Techniques Employed.

Rotation Augmentation. In the project, rotation augmentation is implemented following the approach described by (Shete et al., 2021). Rotation augmentation randomly rotates images within a specified range of -10 to +10 degrees. This technique enhances the model's ability to generalize across different orientations of objects, improving its robustness to variations in image orientation.

Zoom Augmentation. By randomly zooming in or out on images, zoom augmentation replicates changes in scale or distance between the camera and the subject. This method is effective in handling variations in the size of objects within images, such as lesions in dermatoscopic images, and helps the model learn to recognize and classify objects regardless of their size or scale (Shete et al., 2021).

Horizontal Flip Augmentation. Creating mirror images of the original data by randomly flipping images horizontally, horizontal flip augmentation increases the diversity of the dataset. This technique is particularly beneficial for improving the model's ability to generalize across different orientations of objects, making it more adept at recognizing and classifying objects irrespective of their orientation.

Width Shift Augmentation. This technique randomly shifts images horizontally by enhancing the model's resilience to changes in the spatial arrangement of objects. Displaces objects along the horizontal axis, allows the model to learn robust features irrespective of the object's horizontal position in the image frame (Shete et al., 2021).

Height Shift Augmentation. It randomly shifts images vertically, introducing vertical displacements of objects within the image. This technique mimics variations in the positioning of objects, such as lesions in skin cancer images, and helps the model learn to identify features regardless of the object's vertical position (Shete et al., 2021).

By applying data augmentation techniques, the samples of minority classes are increased significantly. Distribution of classes after data augmentation as follows: Actinic keratoses (akiec) with 5118, Basal cell carcinoma (bcc) with 5421, Benign keratosis-like lesions (bkl) with 5800,

Dermatofibroma (df) with 4202, Melanoma (mel) with 5805, Melanocytic nevi (nv) with 6705, and Vascular lesions (vasc) with 4775. This is illustrated in Figure 33.

Figure 33

Skin Cancer Lesion Images Classes After Augmentation



After the completion of data augmentation, the generated images and the metadata are appended to the existing csv file for future steps. This is illustrated in Figure 34.

Figure 34

Dataset After Data Augmentation

	lesion_id	image_id	skin_cancer_type	dx_type	age	sex	\
37821	HAM_0000118	ISIC_0035343		2 histo	72	1	
37822	HAM_0000118	ISIC_0032227		2 histo	80	1	
37823	HAM_0002730	ISIC_0031682		2 histo	80	1	
37824	HAM_0002730	ISIC_0030459		2 histo	75	1	
37825	HAM_0001466	ISIC_0029284		2 histo	70	1	
			localization		clean_image_path		
37821			ear		HAM10000_images_clean/ISIC_0035343		
37822			scalp		HAM10000_images_clean/ISIC_0032227		
37823			scalp		HAM10000_images_clean/ISIC_0031682		
37824			scalp		HAM10000_images_clean/ISIC_0030459		
37825			scalp		HAM10000_images_clean/ISIC_0029284		

This combined data frame consists of newly generated images and their metadata and which is used for the future preprocessing steps.

Principal Component Analysis

After data augmentation implementation, dimensionality reduction technique Principal Component Analysis (PCA) was utilized to reduce the dataset into a smaller dimensional subspace, which lowers the computational and storage needs while preserving the critical features that capture most of the dataset's statistical information (Elgamal,2013).

Created a custom function “list_blobs_with_prefix” to fetch all the image paths with specified prefix. Each image is decoded using “cv2.imdecode” and then flattened into a one-dimensional array. Data is scaled using ‘MinMaxscaler’ to ensure that the pixel values are within a normalized range of 0 to 255.

Calculated full PCA on images to calculate the explained variance ratio. With the help of explained variance ratio, the minimum number of components needed to explain at least 95% of the variance was found. 79 components are required to retain 95% variance as illustrated in Figure 35.

Figure 35

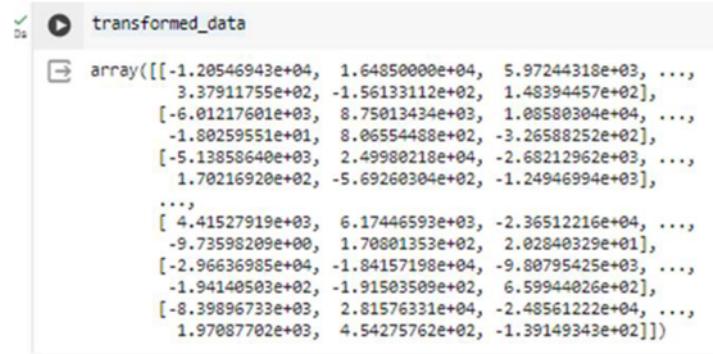
Number of Components to Retain 95% Variance

 Number of components to retain 95% variance: 79

Incremental CA was applied to the batch of the images. To effectively compute the PCA, images are buffered and processed in chunks to ensure that each ‘partial_fit’ call has sufficient data. The image data was processed in batches, and the batches were loaded, preprocessed, added to the buffer and PCA was applied. This process significantly reduces the components to 79, by capturing the variance of 95%. Figure 36 illustrates how the transformed data looks.

Figure 36

Array of Principal Components

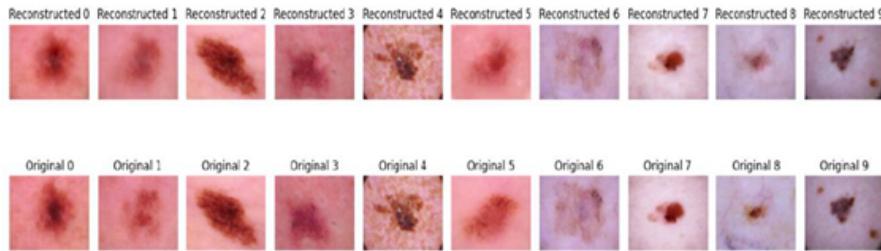


```
transformed_data
array([[-1.20546943e+04,  1.64850000e+04,  5.97244318e+03, ...,
       3.37911755e+02, -1.56133112e+02,  1.48394457e+02], ...,
      [-6.01217601e+03,  8.75013434e+03,  1.06580304e+04, ...,
       -1.80259551e+01,  8.06554488e+02, -3.26588252e+02], ...,
      [-5.13858648e+03,  2.49980218e+04, -2.68212962e+03, ...,
       1.70216920e+02, -5.69260304e+02, -1.24946994e+03], ...,
      [ 4.41527919e+03,  6.17446593e+03, -2.36512216e+04, ...,
       -9.73598209e+00,  1.70801353e+02,  2.02840329e+01], ...,
      [-2.96636985e+04, -1.84157198e+04, -9.80795425e+03, ...,
       -1.94148503e+02, -1.91583589e+02,  6.59944026e+02], ...,
      [-8.39896733e+03,  2.81576331e+04, -2.48561222e+04, ...,
       1.97087702e+03,  4.54275762e+02, -1.39149343e+02]])
```

Where each row represents an image transformed to a new feature space using principal components. The numbers in the array represent the coordinates of the original image in principal component space. Reconstructed the images from PCA components to compare them with the original images. This is illustrated in Figure 37.

Figure 37

Original vs Reconstructed Images From PCA



PCA has helped to reduce the dataset's size without losing important information (Elgamal,2013).

L2 Regularization

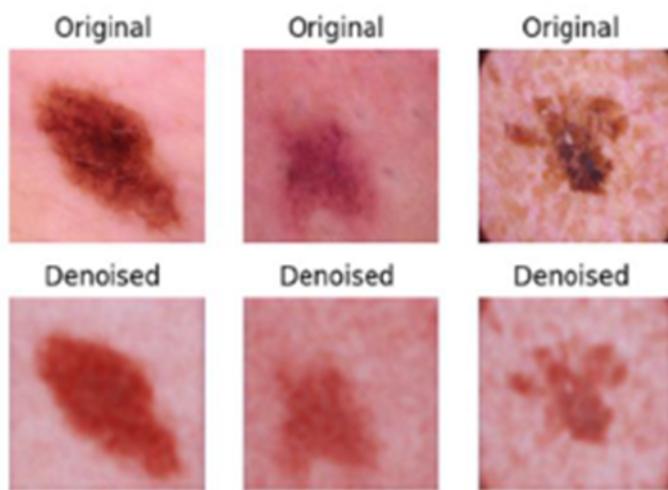
In the next step of preparing the data, L2 regularization was implemented. Regularization techniques are very important in machine learning. They help in avoiding overfitting, leading to where

the model performs poorly on unseen data. Regularization is like adding rules or restrictions to the model, making it learn in a simpler and generalized way (Medhat et al., 2022).

L2 regularization is also called Ridge Regression. In this project, the denoising model is a deep convolutional autoencoder, which includes layers for encoding and decoding. L2 Regularization is employed by penalizing large weights. Adam optimizer with a learning rate 0.01 is used and mean squared error is used as a loss function. Training is conducted with early stopping to prevent overfitting on training data. Below figure compares the original images and images after L2 regularization.

Figure 38

Original Images vs Images After L2 Regularization



The denoised images, processed by the autoencoder, exhibit a smoother appearance, with less random variation and a more homogenized presentation of the lesion features. The denoising process worked well on different types of skin lesions and various levels of severity, showing that the model can effectively handle a wide range of skin cancer appearances.

Data Preparation

The next step in Data Preparation is making the data ready for the modeling phase. To ensure robustness and generalization of models that perform well not only on the training data but also on the

new unseen data, the dataset is split into training, validation, and testing sets. This split enables our model's trained and evaluated on different data which ultimately helps in preventing overfitting and improving the generalization capability to new data.

Training set

The training dataset is used to train the model. The model learns from this data and makes predictions. This is typically 60-80% of the entire dataset. This project took 60% instances of the total data as illustrated in Figure 39 for training.

Figure 39

Training dataset

Validation set

The validation dataset is used to fine-tune the hyper parameters and evaluate the model. This helps in preventing overfitting of the model to the training data. This helps in selecting the best version of the model. This project took 20% instances of the total data as illustrated in Figure 40 for validating the models.

Figure 40

Validation dataset

Testing set

The testing dataset evaluates the model's performance after training and validation. Unlike the validation dataset, the test dataset remains unseen and unused during the training and validation phases. Therefore, it solely serves as a benchmark for evaluating the final model's performance.

This testing dataset gives a realistic understanding of how well the trained and fine-tuned model will perform in real-time. This project took 20% instances of the total data as illustrated in Figure 41 for testing the models.

Figure 41

Testing dataset

age	sex	clean_image_path	localization_abdomen	localization_acral	localization_back	localization_chest	localization_ear	localization_face	localization_foot	localization_genital	localization_hand
3143	0.5625	0 HAM10000_images_clean\ISIC_0025432.jpg	False	False	False	False	False	False	False	False	False
2036	0.9375	1 HAM10000_images_clean\ISIC_0029454.jpg	False	False	True	False	False	False	False	False	False
8083	0.3125	1 HAM10000_images_clean\ISIC_0029730.jpg	False	False	False	False	False	False	False	False	False
6489	0.8125	1 HAM10000_images_clean\ISIC_0027779.jpg	True	False	False	False	False	False	False	False	False
9154	0.4375	1 HAM10000_images_clean\ISIC_0030210.jpg	False	False	True	False	False	False	False	False	False

By preparing the training, validation, testing datasets, the groundwork for the next stages of the project has been laid. These datasets are used to train, fine-tune, and test the deep learning models such as Convolutional Neural Networks, Convolutional Neural Network – Ove-vs-All and the transfer learning models such as ResNet50, DenseNet121.

Data Statistics

This section outlines the results of data preparation, which include converting raw datasets to pre-processed and prepared datasets. The findings are displayed using statistical visualization tools to help understand how the data has developed during the preparation process.

Raw Dataset

The HAM10000 dataset comprises 10015 unique values of images, 7470 unique values of 'lesion_id' were collected from different populations for over a period of 20 years from two sites, the

Department of Dermatology at the Medical University of Vienna, Austria, and the skin cancer practice of Cliff Rosendahl in Queensland, Australia (Tschandl P. et al. 2018).

Figure 42 illustrates the Raw Data Frame, a Python structure with a Range Index that spans 10015 elements from 0 to 10014.

Figure 42

Raw Data Frame

```
> <class 'pandas.core.frame.DataFrame'>
RangeIndex: 10015 entries, 0 to 10014
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   lesion_id        10015 non-null   object  
 1   image_id         10015 non-null   object  
 2   dx               10015 non-null   object  
 3   dx_type          10015 non-null   object  
 4   age              9958 non-null   float64 
 5   sex              10015 non-null   object  
 6   localization     10015 non-null   object  
dtypes: float64(1), object(6)
memory usage: 547.8+ KB
```

The dataset comprises seven columns, all containing non-null entries, indicating that the dataset is fully populated. These columns are labeled as 'lesion_id', 'image_id', 'dx', 'dx_type', 'age', 'sex', and 'localization', all stored as object data types, which are commonly used for textual data in pandas, except for 'age', which is stored as float64 data type.

This arrangement occupies around 547.8+ KB of memory, indicating a significant amount of data. The DataFrame provides essential information for further data manipulation and analysis tasks, giving insights into the dataset's structure and the characteristics of the stored variables.

Pre-processed Datasets

After we've gone through the process of preparing the dataset, including handling any missing values that might have been present, it's evident that now every column contains a total of 10,015

entries with no missing information. This means that any gaps or inconsistencies in the original dataset have been taken care of, ensuring that we have a complete and comprehensive set of data across all columns.

Having this consistency across all columns is a reassuring sign. It suggests that we've successfully dealt with any potential issues that could have affected our analysis. With every column now containing complete information, we can proceed with more confidence in our further analysis, knowing that our dataset is reliable and ready for in-depth examination. This outcome underscores the importance and effectiveness of the steps we've taken to ensure the quality and completeness of our data.

Figure 43

Pre-Processed HAM10000 Dataset

→	lesion_id	10015
	image_id	10015
	dx	10015
	dx_type	10015
	age	10015
	sex	10015
	localization	10015
	dtype:	int64

Transformed Dataset

Initially, the dataset showed an evident disparity in sample distribution across skin cancer Classification. The original counts were 327 instances of Actinic keratoses (akiec) across all columns, 514 instances of Basal cell carcinoma (bcc), 1099 instances of Benign keratosis-like lesions (bkl), 115

instances of Dermatofibroma (df), 1113 instances of Melanoma (mel), 6705 instances of Melanocytic nevi (nv), and 142 instances of Vascular lesions (vasc). These counts provide an initial understanding of the distribution of different skin lesion types in the dataset before any augmentation. After applying data augmentation techniques, there was a substantial increase in the number of samples. This approach successfully addressed the imbalance issue in the dataset by ensuring a more equitable representation of lesions. Prior to augmentation, there were a total of 10015 images present in the dataset. However, after implementing augmentation techniques it expanded to contain a total of 37826 samples which significantly increased available data volume as shown in Figure 44.

Figure 44

Transformed Dataset Comparison

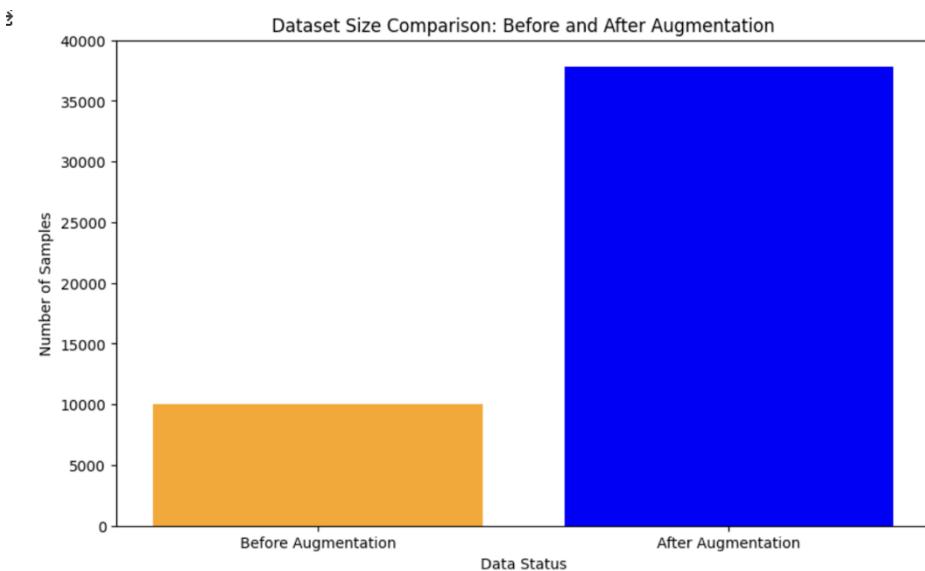
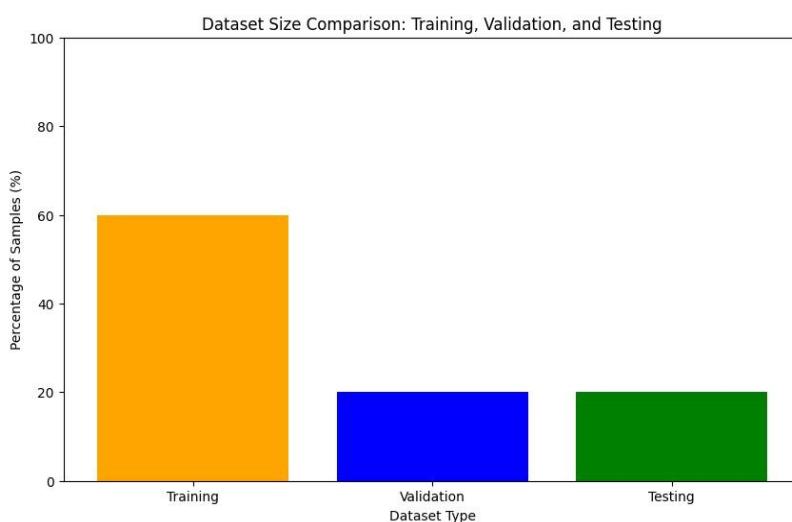


Figure 45*Transformed Dataset*

	lesion_id	image_id	skin_cancer_type	dx_type	age	sex	\
37821	HAM_0000118	ISIC_0035343		2	histo	72	1
37822	HAM_0000118	ISIC_0032227		2	histo	80	1
37823	HAM_0002730	ISIC_0031682		2	histo	80	1
37824	HAM_0002730	ISIC_0030459		2	histo	75	1
37825	HAM_0001466	ISIC_0029284		2	histo	70	1
	localization			clean_image_path			
37821	ear	HAM10000_images_clean/ISIC_0035343					
37822	scalp	HAM10000_images_clean/ISIC_0032227					
37823	scalp	HAM10000_images_clean/ISIC_0031682					
37824	scalp	HAM10000_images_clean/ISIC_0030459					
37825	scalp	HAM10000_images_clean/ISIC_0029284					

Prepared Dataset

The dataset was divided into training, validation, and test sets with sample sizes of 60%, 20% and 20% of the dataset. Figure 46 provides a visual representation of the dataset distribution with a specific allocation for training, validation, and testing subsets.

Figure 46*Transformed Dataset Comparison*

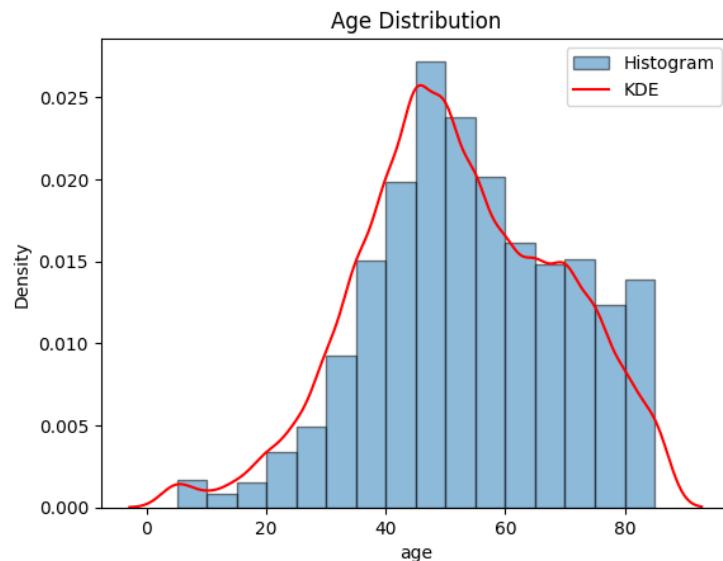
The bar chart visually represents the division of a dataset into training, validation, and test sets. The training set, shown in cornflower blue, accounts for 60% of the data, reflecting its primary role in model development where the algorithm learns and identifies patterns from the data. The validation and test sets are equally sized, each comprising 20% of the dataset. The validation set, represented in medium sea green, is used for fine-tuning the model parameters and provides an unbiased evaluation during the training phase. The test set, depicted in salmon, is critical for the final evaluation, where the model's performance is assessed on new, unseen data to determine its predictive accuracy and generalization capabilities in real-world scenarios.

Data Analytics Results

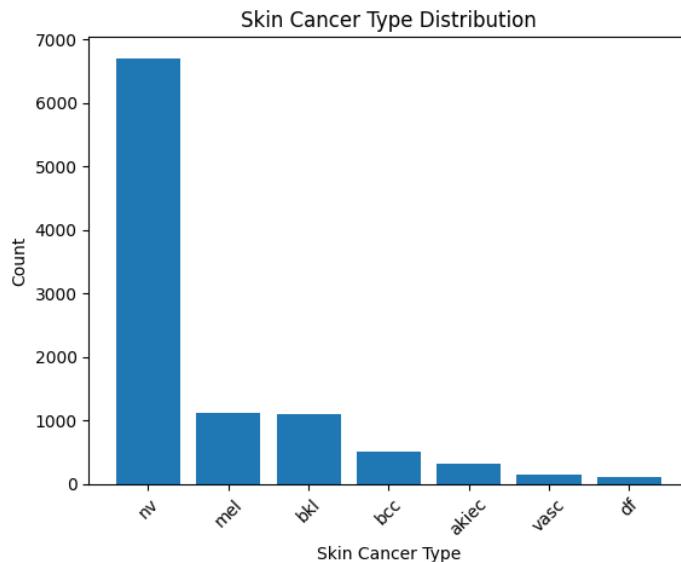
The results of diverse data analytics performed on HAM10000 metadata using big data visualization formats provided valuable insights into the Skin Cancer Classification Project.

Age Distribution

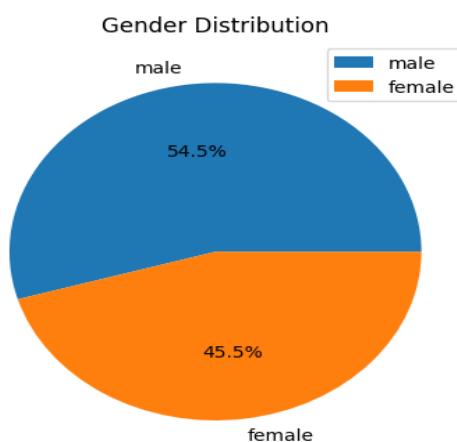
It is essential to understand the distribution of the ages in the dataset to effectively classify the skin cancer lesions. Before age imputation, ages in the HAM10000 dataset ranged from 0 to 85 years, while after imputation, they ranged from 5 to 85 years, as observed from age distribution in Figure 33. It is observed that patients are generally older, and the age distribution appears to be a bimodal distribution, with concentration of patients aged between 35 to 55 and 60 to 75. This pattern aligns with the typical demographic distribution of skin cancer, which primarily affects older individuals.

Figure 47*Age Distribution**Cancer Type Distribution*

The objective of this project is to classify the skin cancer type. So, it is essential to understand the distributions of skin cancer types in our dataset. The bar chart in Figure 34 reveals a significant imbalance in the HAM10000 dataset; the 'nv' category alone comprises 6705 images, outnumbering the combined total of all other categories. This imbalance in the dataset poses challenges for the development of machine learning algorithms. To mitigate this, we performed data augmentation to generate images of less represented categories.

Figure 48*Skin Cancer Type Distribution****Gender distribution***

Gender can influence the skin cancer patterns, which is why it is very important to understand the Gender distribution of the dataset. This distribution in Figure 49 revealed that the male category has more representation than the female category by 10%. This could either mean that the dataset is skewed towards males or skin cancers typically appear more among the male populations.

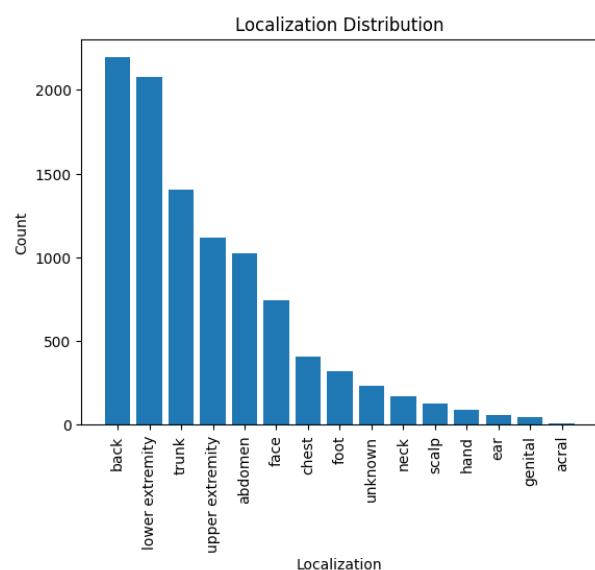
Figure 49*Gender Distribution*

Localization Distribution

It is very important to gain an understanding of the localizations of skin cancer lesions as they potentially throw some light on the skin cancer type as the lesions vary depending on the cancer type and the location of it. The bar chart in Figure 50 shows that the localization of skin cancer lesions are heavily imbalanced in the HAM10000 dataset. Most of the lesions are concentrated on the back, lower extremity.

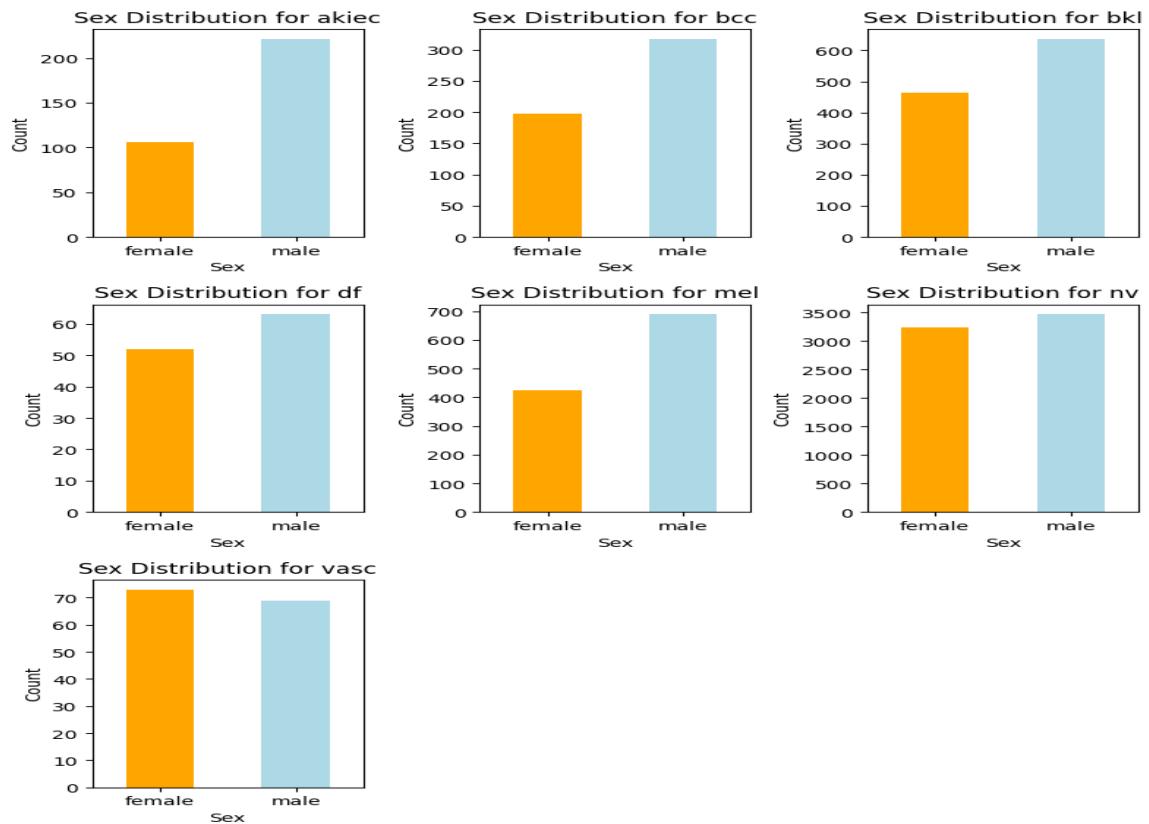
Figure 50

Localization Distribution

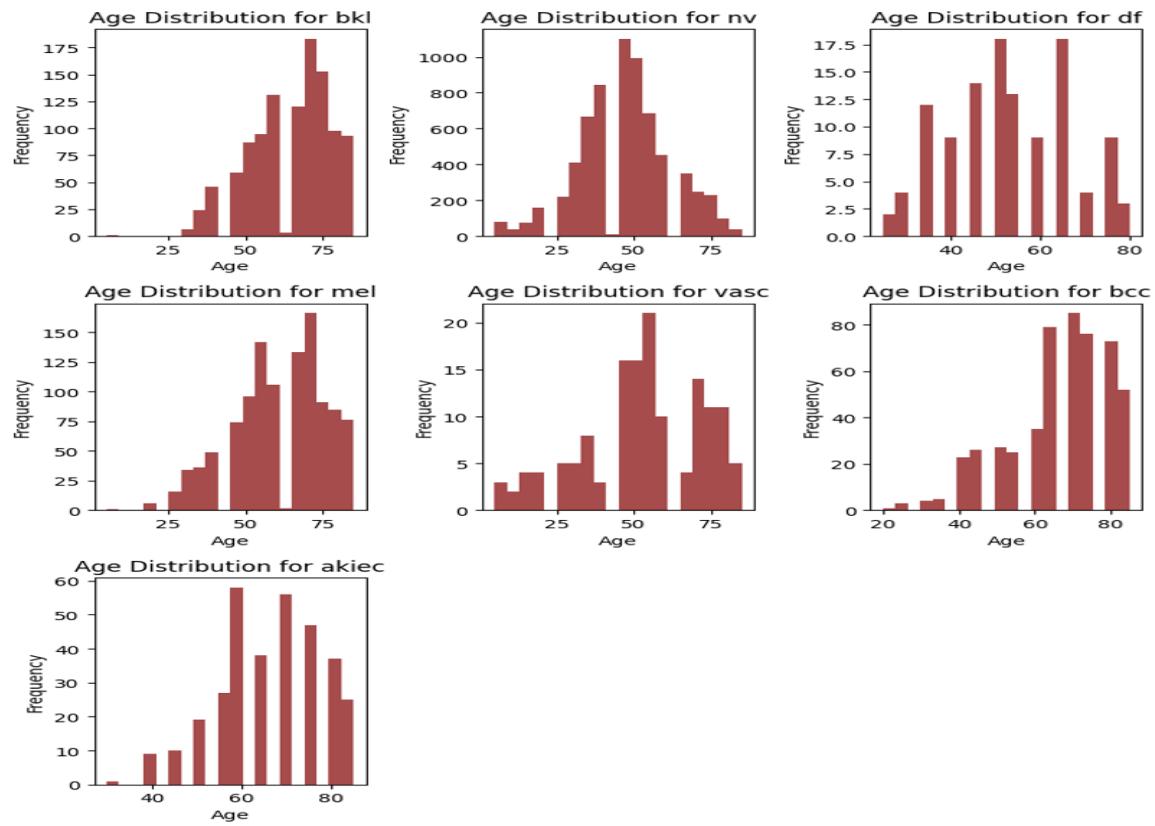


Gender distribution by skin cancer type

In order to gain a better understanding of the gender distribution across the cancers, plotted gender distribution by skin cancer type as illustrated in Figure 51. From this Figure 51 it has been observed that some skin cancer types like 'akiec', 'bcc', 'bkl', and 'mel' are more prominent among the males compared to females. This suggests that these cancers might occur more often in men than in women. However, we cannot conclude this inference because this skewness could be due to the factor that there are more skin cancer lesion examples of men in the dataset. For cancer type 'vasc' it is observed that women were affected more than men.

Figure 51*Cancer Type wise Gender Distribution**Age distribution across all cancer types*

In order to gain a deeper understanding of the age distribution across the cancers, plotted age distribution by skin cancer type as illustrated in Figure 52. From this Figure 52 it is observed that in general skin cancers are more prominent among the older age patients but however it has been observed that certain cancer types, such as 'nv', 'vasc' and 'df' affect patients of ages below 30.

Figure 52*Cancer Type wise Age Distribution*

Model Development

Model Proposals

The objective of the project is to fine-tune and evaluate the DenseNet121 deep learning model for accurate classification of skin lesion images into different classes of skin cancer. The models were trained and evaluated on the HAM1000 dataset, acquired from Harvard Dataverse, which aims to represent a broad range of skin cancer cases with high-quality images. The cleaned and transformed dataset is then modeled and evaluated using CNN, CNN-OVA, ResNet50, and DenseNet121.

Convolutional Neural Networks

Literature Survey. Agarwal and Singh (2022) emphasized on Convolutional Neural Networks for skin cancer classification due to its deep learning capabilities and its capacity to handle complex data variability present in skin lesion images. They employed data augmentation techniques such as random flipping, zooming and used advanced activation functions such as ReLU and Leaky ReLU to introduce non-linearity in the model. This helped in learning more complex patterns in the images. Batch normalization, flattening layers, pooling layers, and dropout layers were included to increase the rate of convergence of the network and to prevent overfitting, which enables the model to perform well on new unseen data. The study compares various CNN architectures and their performances to aid dermatologists in their predictions.

Fuadah et al. (2020) highlights the effectiveness of Convolutional Neural Networks (CNNs) in image analysis, leveraging their architecture to extract complex patterns in 2D data. The study mentions that CNNs are deep neural networks that contain several hidden layers, like convolutional and pooling layers for feature extraction, ReLU activation function for introducing complexity, and fully connected layers for classification. CNNs also use advanced optimizers like Adam and Nadam for faster convergence and for more accurate classifications. CNN models in skin cancer classification have the ability to surpass

dermatologists in classifying different skin cancers from lesion images, making it particularly valuable to regions with limited access to dermatologists.

Dildar et al. (2021) conducted a thorough analysis of deep learning methodologies, particularly focusing on Convolutional Neural Networks (CNNs) and Artificial Neural Networks (ANNs), for the detecting skin cancers. This study also mentions data preprocessing and feature extraction steps, and the detailed architecture employed by neural networks for detection. Their proposed CNN architecture contains several hidden layers including convolution layers, pooling layers and three fully connected layers. This research highlights the potential of these models in healthcare sector with significant contributions to the field of dermatological diagnostics especially in diagnosing skin cancers.

Brinker et al. (2018) offers an overview of the effectiveness of Convolutional Neural Networks (CNNs) in the classification of skin lesions. The study presents the first systematic review focused on CNNs employed specifically for skin cancer lesion classification. It highlights the difference between a regular neural network and a convolutional neural network, which is the way in which the neurons are connected to their previous layers. The study examines two approaches for categorizing the skin lesions, an end-to-end CNN model created from scratch and a pre-trained CNN architecture, such as ImageNet, that uses transfer learning approach. The paper emphasizes that CNNs are state-of-the-art classifiers for skin lesions.

Fabio et al. (2018) presents a comprehensive analysis of various Convolutional Neural Network (CNN) architectures in classifying melanoma. The paper evaluates nine different CNN architectures across multiple configurations over five dataset splits, where each dataset is tested thrice, making it a total of 135 models. It was observed that there is a weak correlation between CNNs performance in general image recognition tasks (ImageNet) when compared to their effectiveness in a specific classification task like melanoma classification. This suggests that the performance of the model greatly

depends on the dataset variability and randomness. The study also recommends using ensembles of models rather than solo models for better classification results.

Convolutional Neural Network Architecture. A Convolutional Neural Network (CNN) is a deep learning algorithm that is designed specifically to process two-dimensional data like image data, and audio data (Agarwal & Singh, 2022). It is primarily used for image detection and classification problems, and in the current scenario it is being used to categorize skin cancer lesions.

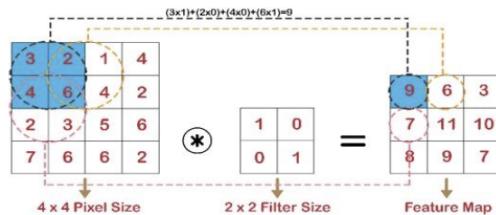
The architecture of a CNN typically consists of several layers including Input Layer, Convolutional Layer, Pooling Layer, Activation Functions, Flattening Layer, Fully Connected Layers, and Output Layer.

Input Layer. This is the first layer of a CNN architecture, which typically receives the raw data in the form of a two-dimensional image. The image data is typically represented in two dimensions: height and width and in some cases, depth is also mentioned. The latter typically represents the number of color channels, one for grayscale images and three for color images.

Convolutional Layer. This is the basic building block of a CNN algorithm. These layers process the image data by parsing the images with various kernels or filters to extract relevant features from the image, as discussed in Fuadah et al. (2020). The output of each filter is typically a feature map that is a result of the convolution done by the kernels on the images. Figure 53 shows the illustration of this convolution process.

Figure 53

Convolution Process Illustration



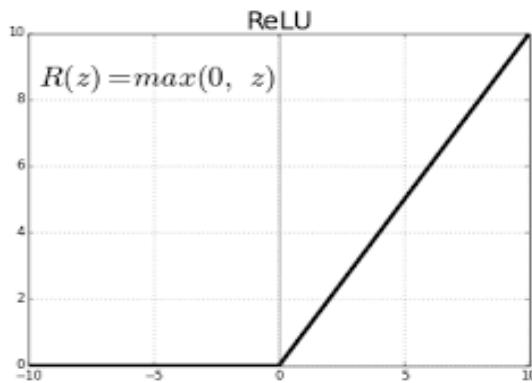
Note. This image is taken from Fuadah et al. (2020)

Activation Layer. After convolution, activation functions are applied to introduce non-linearity in the model (Agarwal & Singh, 2022). This enables the model to learn complex patterns in the data. Most common activation functions are Rectified Linear Unit (ReLU) and SoftMax.

ReLU. It is a widely used activation function which is typically an identity function of positive inputs and a zero for negative inputs, as shown in Figure 54.

Figure 54

ReLU Activation Function



Note. This image is taken from Sunarya et al. (2023)

SoftMax. This activation function is generally used in the final stage, after the output layer. It transforms the raw outputs to probabilities of class labels (Agarwal & Singh, 2020), especially for a multiclass classification problem, like the skin cancer lesion classification. Equation 1 gives the formula for SoftMax activation function, which is taken from Agarwal and Singh (2020).

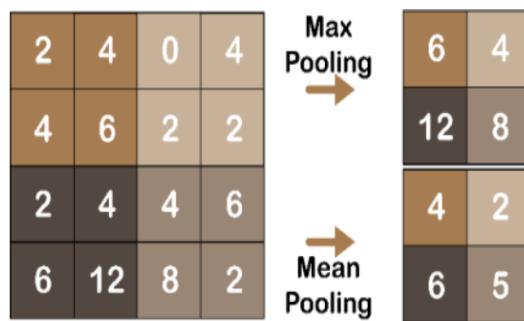
$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (1)$$

Pooling Layer. This layer essentially down samples the feature map that it receives from the convolutional layer, to reduce the number of dimensions and the number of parameters for the next convolutional layer (Fuadah et al., 2020). Thus, it performs dimensionality reduction which brings down

the computational cost of the model and improves the generalization of the model. Max pooling and Average Pooling are the most common pooling layers used in CNN models. Figure 55 illustrates an example of max and average pooling layers.

Figure 55

Pooling Layers



Note. This image is taken from a paper by Fuadah et al. (2020)

Flattening Layer. These layers act as a bridge between the convolutional layers and fully connected layers by preparing the data for the latter. They flatten the input 3D data that it receives into a 1D vector/array that is fed into fully connected layers (Fuadah et al., 2020).

Fully Connected Layer. These are the dense layers in a neural network that connects every input neuron to every other output neuron. These are usually employed at the end of architecture (Dildar et al., 2021) that are followed generally by a SoftMax activation function if it is a multiclass classification problem or a sigmoid activation function for binary classification.

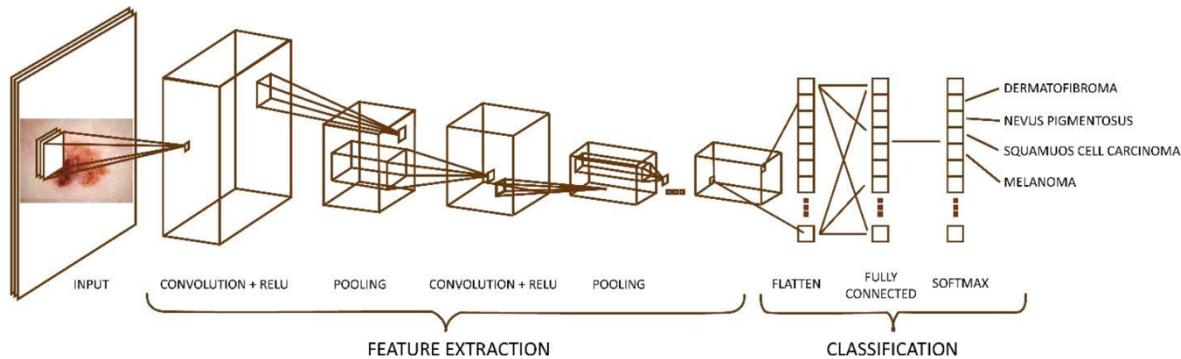
Output Layer. This is the last layer in the model that produces the final output, either probability distribution values, by using a SoftMax activation function for classification tasks or a regression value for regression tasks.

Apart from these standard layers of a CNN model, Dropout layers were employed for skin cancer lesion classification CNN model. Dropout Layer is a data regularization technique that drops neurons at

random to prevent overfitting and thereby generalizing the model (Agarwal & Singh, 2020). Figure 56 illustrates the CNN architecture used for CNN modeling and Table 19 details the pseudo-code of this model.

Figure 56

CNN Architecture



Note. This image is taken from a paper by Fuadah et al. (2020)

Table 19

Pseudo Code for CNN Model

Model Layers	Layers Description
Input Layer	Takes the input image data, which is a PCA transformed image.
Conv2D Layer	Applies 32 convolutional filters of size (3, 3) with ReLU activation function and L2 regularization to the image data.
Conv2D Layer	Applies 64 convolutional filters of size (3, 3) with ReLU activation function and L2 regularization.
AveragePooling2D Layer	Down samples the features extracted and retains the most important features thereby reduces the computational load and prevents overfitting.

Model Layers	Layers Description
Dropout Layer	Applies a dropout rate of 0.5 to prevent overfitting.
Conv2D Layer	Applies 128 convolutional filters of size (3, 3) with ReLU activation function and L2 regularization.
Conv2D Layer	Applies 128 convolutional filters of size (3, 3) with ReLU activation function and L2 regularization.
AveragePooling2D Layer	Down samples the features extracted and retains the most important features.
Dropout Layer	Applies a dropout rate of 0.5 to prevent overfitting.
Flatten Layer	Flattens output from convolutional layers to a 1D vector.
Fully Connected Layer	Fully connected layer with 128 neurons and ReLU activation function, with L2 regularization.
Dropout Layer	Applies a dropout rate of 0.5 to prevent overfitting.
Flatten Layer	Flattens output from convolutional layers to a 1D vector.
Fully Connected Layer	Fully connected layer with 128 neurons and ReLU activation function, with L2 regularization.
Dropout Layer	Applies a dropout rate of 0.5 to prevent overfitting.
Input Layer	Takes the input metadata.

Model Layers	Layers Description
Fully Connected Layer	Fully connected layer with 256 neurons and ReLU activation function, with L2 regularization.
Dropout Layer	Applies a dropout rate of 0.5 to prevent overfitting.
Fully Connected Layer	Fully connected layer with 128 neurons and ReLU activation function, with L2 regularization.
Dropout Layer	Applies a dropout rate of 0.5 to prevent overfitting.
Concatenate Layer	Combines the outputs of the image branch and the metadata branch.
Fully Connected Layer	Fully connected layer with 256 neurons and ReLU activation function, with L2 regularization.
Dropout Layer	Applies a dropout rate of 0.5 to prevent overfitting.
Fully Connected Layer	Fully connected layer with 128 neurons and ReLU activation function, with L2 regularization.
Dropout Layer	Applies a dropout rate of 0.5 to prevent overfitting.
Output Layer	SoftMax activation function converts the raw outputs of the fully connected layer to probabilities for each class of skin cancer types.
Training	Adjusting the number of epochs, batch size, learning rate based on training and validation accuracies.
Evaluation	Evaluate model's performance over a range of metrics on the test dataset

Model Optimization. CNN models can be optimized by fine tuning hyper parameters, augmenting data, and by data regularization. Hyperparameter tuning involves adjusting the number of layers i.e., the depth of the model, adjusting the number of neurons which is the width of each layer and adjusting the learning rate for faster convergence of the optimizer, and adjusting the number of epochs to avoid overfitting or underfitting and adjusting the batch size for optimal efficiency. Data augmentation included zooming, rotating, and flipping of images for better generalization capability of the model (Syafa'ah et al., 2021). Principal Component analysis (PCA) was performed to reduce the dimensionality of the data for reducing the computational load and other data regularization techniques such as adding Dropout Layers were applied to prevent overfitting by dropping neurons randomly in hidden layers and L2 Regularization, which adds a penalty term to the loss function is used for better generalization of the model. These model optimization techniques improved the performance of the model by reducing overfitting and making generalized predictions on new and unseen data.

Convolutional Neural Network - One-Versus-All (CNN-OVA)

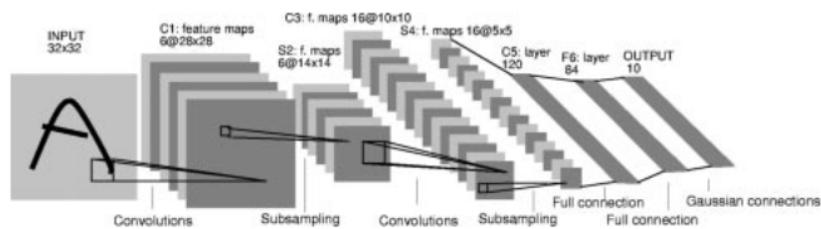
Literature and Technology Survey. The Convolutional Neural Network-One-Versus-All model is a special version of the usual CNN design. The classification performance significantly improved by combining these two strategies, going from 77% accuracy to 92.09% accuracy (Polat & Koc, 2020).

In 1950s and 1960s, Hubel and Wiesel identified distinct neurons in the cat's visual cortices reacting separately to the visual fields. This foundational research influenced the design of neural networks (Hubel & Wiesel, 1968). Kunihiko Fukushima (1980) advanced this research by developing neocognitron, which includes convolutional and down sampling layers. Significant advancement was made for automated image recognition (LeCun et al., 1989), by utilizing backpropagation to train CNNs for automatic feature extraction from images. In the 2000s, using GPUs (Graphics Processing Units) to run CNNs (Convolutional Neural Networks) made them run much faster and this improvement advanced

the fields of image processing and machine learning (Krizhevsky et al., 2012). Figure 57 is a detailed view of CNN architecture using back propagation for recognizing the digits.

Figure 57

CNN Model Architecture

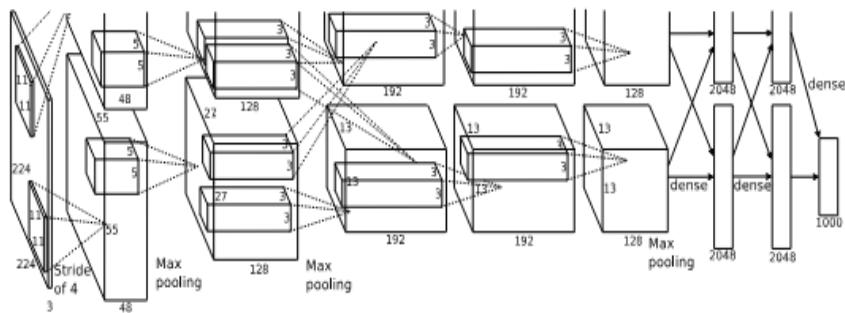


Note. The architecture is taken from the work by LeCun et al. (1998)

Figure 58 represents the communication between GPUs in the layers.

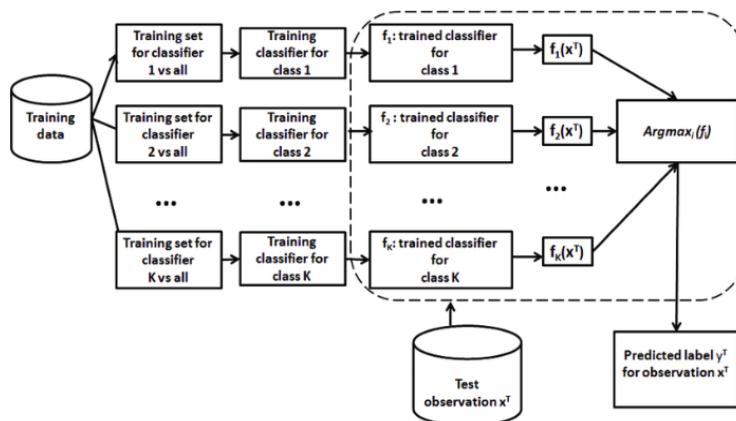
Figure 58

Architecture of CNN Using GPUs



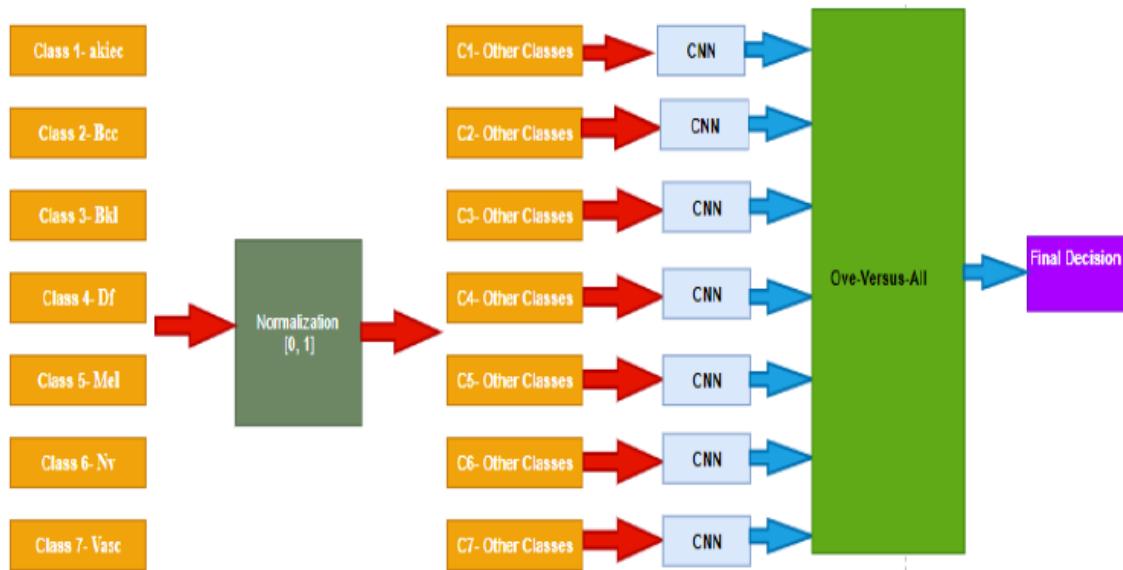
Note. This image is taken from the work done by Krizhevsky et al. (2012)

One-vs-All (OvA) is a machine learning method used for multi-class classification. For each class, OvA creates a binary classifier, and this classifier determines whether the instance belongs to the respective class or other classes. Figure 59 illustrates the One-vs-All (OvA) architecture.

Figure 59*One-vs-All Architecture*

Note. The architecture is taken from the work done by Escalante et al. (2013)

To improve the classification performance, CNN and OVA techniques are combined (Polat & Koc, 2020) and leading to an increase in accuracy from 77% to 92.09%. Figure 60 explains the integration of CNN and OVA approaches.

Figure 60*CNN – OVA Architecture*

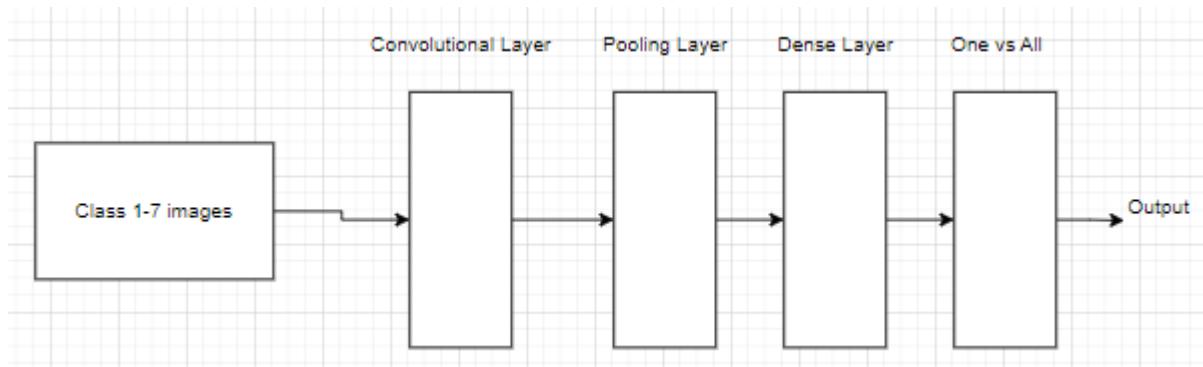
Note. This architecture is taken from the work done by Kemal Polat and Kaan Onur Koc (2020)

Initially, all the classes undergo normalization to scale the data between 0 and 1. By using CNN-OVA, each class is identified by a separate model. Each model checks whether the data belongs to their class or any other class. There are seven models for seven classes. The results from all these models are then analyzed to make a final decision.

CNN-OVA Model Architecture. There are seven different classes, which represents images are classified based on the class. Then the images are given as input to the input layer, which is the first layer of architecture. And it is responsible for receiving the input images. After the images are received by the input layer, then they are fed into convolutional layer where the features of the images are learnt by the layer. In the next stage images are given to the pooling layer, this layer basically reduces the dimensions of the images by keeping the most important information. After extracting the features and reducing the dimensions of images then the images are fed into dense layer, by using the learned rate it maps input to the outputs. Then this is integrated with one versus all strategy which effectively classifies the images.

Figure 61

CNN-OVA Model Architecture



Model Optimization. Datta et al. (2021) optimized the deep learning models of skin cancer classification, which involves adding dropout and early stopping regularization to the network. Adjusting

parameters like learning rate, number of layers, and batch size, model can perform even better. These refinements help the CNN-OVA model to learn and generalize effectively from the complex details in dermatoscopic images.

DenseNet121

Literature Review. Nair et al. (2022) proposed a model that works on a deep learning approach using transfer learning from a pre-trained DenseNet-121 architecture for classifying multi-class skin lesions into one of the seven classes used for early diagnosis of skin cancer. In the current research, by the use of transfer learning and Densenet-121, an accuracy of 82.1% was obtained during the training of the model with the HAM10000 dataset, consisting of 10015 dermoscopic images. In fact, the training for any type of classification using a balanced dataset outperforms that of an imbalanced dataset, suggesting that it is necessary to avoid giving preference to classes where a large number of images are available. The study suggests that possible future works include the use of test datasets for evaluating the system performance and the application of advanced data augmentation techniques and generative models toward further balancing class representation.

DenseNet121 provided promising results in classifying the medical images. Chauhan et al. (2021) detected COVID-19 in chest X-ray images in an accurate way, by applying the DenseNet121 architecture within a deep learning framework. The experiment has been detailed and automatic comparisons have been made with the optimizers, loss functions, and learning rate schedulers, in this work, to achieve an accuracy of 98.45% for normal-healthy and 98.32% for COVID-19 images. Further, the performance of DenseNet121 in medical imaging applications has been shown in this work by using transfer learning and early stopping. This will create promising ways for fast and accurate diagnostics of COVID-19.

Neeshma et al. (2022) performed a study on the classification of skin cancer by using pre-trained deep learning architecture, DenseNet-121. They experimented on the imbalanced original dataset and then on the resampled balanced one in this context. Experiments show that when a balanced dataset is

used, there is a significant increase in classification accuracy without any application of additional data preprocessing or augmentation techniques. This work again underscores the significance of having a balanced dataset and proposes the following future research, having the potential to develop novel data augmentation techniques possibly implemented with generative models to have a positive impact on the balance of the dataset.

In a study, Kausar et al. (2021) explored multiclass classification in skin cancer, with a focus on how to utilize DenseNet within deep ensemble learning models. Integrating these diverse learners with DenseNet through majority voting and weighted majority voting drives the ensemble of deep learners to an accuracy of 98% and 98.6%, respectively, better than these learners in individual and the dermatologists' accuracy levels. Promising results, therefore, offer an insight that a DenseNet architecture and an ensemble approach are a combination that is efficient in boosting classification accuracy even for challenging diagnostic tasks.

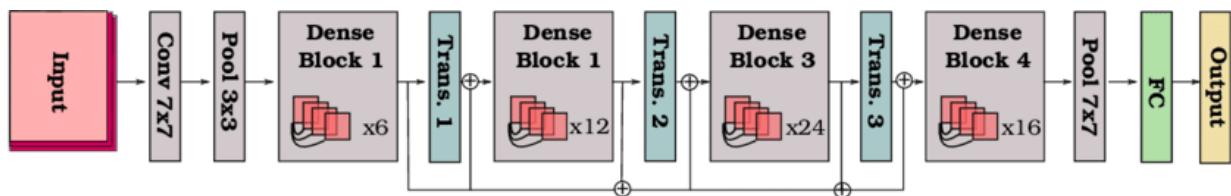
Another relevant application is in the classification of radiographic images. Aldeen (2021) used DenseNet121, a fully connected convolutional neural network, to classify radiographic images into two main classes: normal and abnormal, in order to assist radiologists in their diagnosis. DenseNet121 has achieved high diagnostic accuracy, which had therefore resulted in 87.5%; hence, it has proven very effective for accurate diagnosis. This indicates that the usefulness of the DenseNet121 network in helping radiologists make decisions will increase the diagnostic accuracy in the field of application for radiographic image classification.

DenseNet121 Model Architecture. DenseNet is a modern CNN architecture designed for visual object recognition, known for its state-of-the-art performance with a reduced parameter count. Unlike traditional CNNs, DenseNet employs a unique approach called concatenation, where the output of each layer is utilized to create feature maps and is combined with the next layers. This allows for enhanced feature reuse and propagation throughout the network, contributing to improved learning efficiency and

performance (Hasan et al., 2021). The DenseNet implementation comprises three types of blocks. First is the convolution block, which serves as the fundamental unit of the dense block. Secondly, the dense block is central to DenseNet, where convolution blocks are concatenated and densely connected. Lastly, the transition layer links two adjacent dense blocks. Within the dense block, feature map sizes remain constant, and the transition layer is responsible for reducing the dimensions of the feature map. Additionally, all blocks incorporate the bottleneck design technique (Ji et al., 2019). The architecture of DenseNet121 is depicted in Figure 62.

Figure 62

DenseNet121 Architecture



Note. This is taken from Radwan, N. (2019).

Convolution Layer. A convolution layer within a neural network is a structure that processes input information by using a set of filters (kernels) through which the input, such as an image, slides. It's used to find features in the input and trace them across the input. Each filter produces an output feature map in response to the inputs that select patterns, such as edges or textures, in the input image. This eventually reduces the number of parameters compared to fully connected layers and makes the network efficient in recognizing spatial hierarchies and features across the input.

Pooling Layer. Pooling is the process of down sampling input data to reduce dimensionality, similar to how reducing the resolution of an image would. It keeps the maximum value in each 2x2 sub-region with a stride of 2; hence, it reduces the size of data but keeps the important features, losing spatial details. Variants are possible, for instance, with 3x3 filters with stride 2.

Dense Block. In a DenseNet-121, the numerous convolutional layers within a Dense Block obtain input from all of the previous layers for effective feature reuse and gradient flow. Each layer is followed by a batch normalization and a ReLU activation function to bring non-linearity and often contains a bottleneck layer to make it computationally efficient. The layers collectively add a fixed number of new feature maps, hence the naming of the growth rate, which secures effective feature propagation, reduces the number of overall parameters, and increases the representational power of the network (Albelwi, 2022).

Transition Layer. After each dense block is a transition layer, consisting of convolution and pooling. It does compactness improvements by reducing the number of feature maps (Albelwi, 2022).

Classification Layer. The final layer of DenseNet-121 is a dense fully connected layer that calculates the probability of the different classes of skin lesion images through the softmax activation function. The SoftMax function leads the probabilities to sum up to one, giving a clear and interpretable prediction regarding the classification task, be it on the type or presence of a skin lesion. The equation of SoftMax is as shown in Equation 2, where Z is the input vector.

$$sm(z)_i = \frac{e^{z_i}}{\sum_{j=1}^C e^{z_j}} \quad \text{for } i = 1, \dots, C \quad (2)$$

In Addition to the pre-processed and transformed images, metadata has also been integrated into the model to train the dataset upon. The pseudocode of DenseNet121 model has been summarized in Table 20.

Table 20

Pseudocode of DenseNet121 Model

Layer Type	Description
Input Layer	Represents input data. Two inputs: resized images and metadata. Image shape: (224, 224, 3)
Base DenseNet121 Layer	DenseNet121 pre-trained model without the top layer. Layers are not trainable. Global Average Pooling 2D layer for spatial aggregation
Dense Layer 1	Dense layer with 1024 units, ReLU activation, and L2 regularization. Dropout layer with 0.5 rate for regularization
Metadata Dense Layer 1	Dense layer with 512 units, ReLU activation, and L2 regularization. Dropout layer with 0.5 rate for regularization
Metadata Dense Layer 2	Dense layer with 256 units, ReLU activation, and L2 regularization. Dropout layer with 0.5 rate for regularization
Combination Layer	Combines outputs from DenseNet121 and metadata branches using concatenation
Output Layer	Dense layer with 7 units and softmax activation for multi-class classification

Layer Type	Description
Training	Set up model compilation with Adam optimizer, sparse categorical cross entropy loss, and accuracy metric. Train model with resized image data and metadata. Include validation data for monitoring performance
Evaluation	Evaluate model on the test set. Report test loss and accuracy metrics
Metrics Calculation	Calculate and print F1 score, precision, recall, and confusion matrix using model predictions on the test set

Model Optimization. The pre-trained DenseNet121 serves as the base model for classifying skin cancer, known for its excellent feature extraction capabilities due to its dense connectivity pattern. This architecture captures fine details in medical images effectively. To boost performance and robustness, several optimization strategies were employed. Above the base DenseNet121 architecture, additional dense layers with ReLU activation and dropout layers were integrated into both the image and metadata branches. From the DenseNet121 base, the flow continues through a Global Average Pooling 2D layer, followed by a dense layer with 1024 units featuring ReLU activation and L2 regularization. A dropout rate of 0.5 helps prevent overfitting by randomly deactivating neurons during training.

The metadata branch processes input through two dense layers of 512 and 256 units, each with ReLU activation and L2 regularization. Dropout layers follow each dense layer to enhance regularization throughout the training process. This dual-path setup effectively merges image data with relevant metadata, enriching the feature set.

The combined outputs from these branches are then concatenated and directed through a final dense layer with SoftMax activation. This layer manages the multiclass classification, outputting a probability distribution across the various skin cancer classes. The model is compiled with an Adam optimizer at a learning rate of 0.001, using sparse categorical cross entropy as the loss function, suitable for multiclass tasks. Validation is performed with an independent set to monitor performance and mitigate overfitting.

These collective optimization strategies enhance the model's accuracy and generalizability, ensuring robust performance in skin cancer image classification. The integration of regularization techniques, dropout layers, and the combination of image and metadata features are crucial in achieving a well-balanced and effective model.

ResNet50

Literature Review. Hosny et al. (2020) performed detailed research to identify skin melanoma by utilizing deep convolutional neural networks (DCNNs) like ResNet-101. Their approach includes altering these networks' last three layers in order to enhance the feature extraction capabilities for dermatoscopic images. To address the issues of imbalanced datasets and increase the model adaptability, they utilized region of interest (ROI)-based approaches in addition to the data augmentation. This research found that these modifications significantly enhanced the classification accuracy of these networks, with RESNET showing the most promise of the evaluated models due to its architectural benefits in analyzing complicated images data.

Rezvantalab et al. (2018) conducted extensive research utilizing the ResNet-152, among deep learning architectures, to classify skin cancer lesions from dermatoscopic images. This research found that ResNet-152 excelled better than qualified dermatologists in detecting melanoma and basal cell carcinoma, significantly higher than dermatologists' accuracy rates of 82.26% and 88.82%, respectively. ResNet-152 showed around ROC AUC values of 94.40% for melanoma and 99.30% for basal cell

carcinoma. This research further highlighted the potential for incorporating these advanced neural network models into clinical practice, delivering a tool for enhancing detection and treatment planning. This combination has the opportunity to significantly improve the patient outcomes through providing an honest second opinion while decreasing the error rate in diagnosis associated with the human examination. The research results indicate evolution towards increasingly automated evaluation methods in dermatology, illustrating the importance of deep learning in healthcare imaging.

Razzak et al. (2020) developed multiple stages fully connected deep residual network (ResNet) to categorize skin cancer, by making major enhancements when considering challenging dermatoscopic images. They basically created a very unique design that utilized identity bypass hyperlinks to reduce gradient that dissolve, which is very important for quickly training deeper networks. The findings from experiment results on the ISC- 2018 dataset showed major improvements in classification accuracy, with an incredible 96.07%. This research demonstrates the potential of deep residual networks to improvise the diagnostic performance beyond the existing approaches, making significant contributions to the field of dermatological image analysis.

ResNet Model Architecture. In their research paper on deep residual learning, He et al. (2016) introduced the concept of ResNet models, which mainly solves the issues of training very deep neural networks. They introduced ResNet50, among all other models, which uses residual blocks to assist deep training networks while minimizing the performance decrease that usually comes with greater depth. These residual blocks enable the network to properly learn identity mappings, ensuring that adding more layers does not negatively impact performance but rather improves it, perhaps beating human-level performance on tasks such as ImageNet classification.

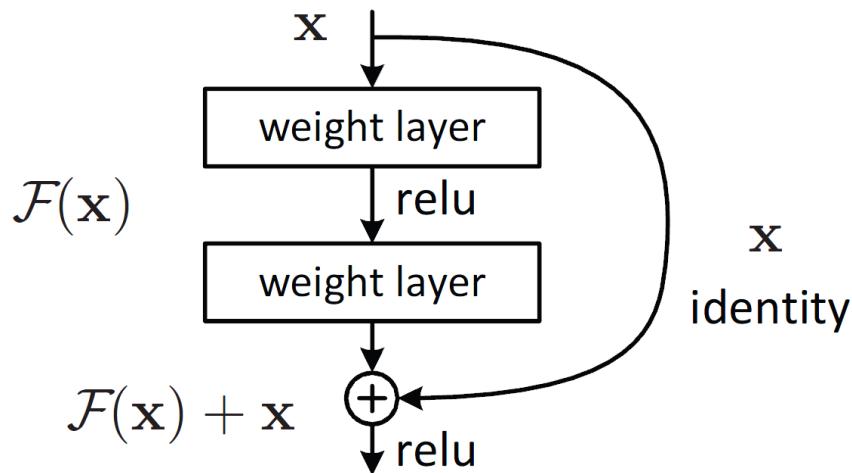
ResNet50, stated by He et al. in 2016, is an innovative neural network architecture meant to tackle the challenges of training deeper networks while minimizing the performance loss. ResNet50's genius resides in the use of shortcut connections, which enables the information to skip particular phases. This

new technique allows the network to properly learn identity functions, overcoming the vanishing gradient problem common in deep learning. So as a result, ResNet50 excels in challenging age identification tasks, and it proved that it is versatile and superior in a variety of deep learning areas.

In Figure 63 the ResNet utilizes a unique design that combines residual blocks with skip connections in order to significantly minimize the issue of gradients disappearing.

Figure 63

Residual Learning Framework:



Note. This figure is taken from the paper by (He et al., 2015)

This setup enables fast training of deeper networks by preserving gradient flow throughout. Each block contains layers where the input is added to the output, which is formulated by equation 3:

$$H(x) = F(x) + x \quad (3)$$

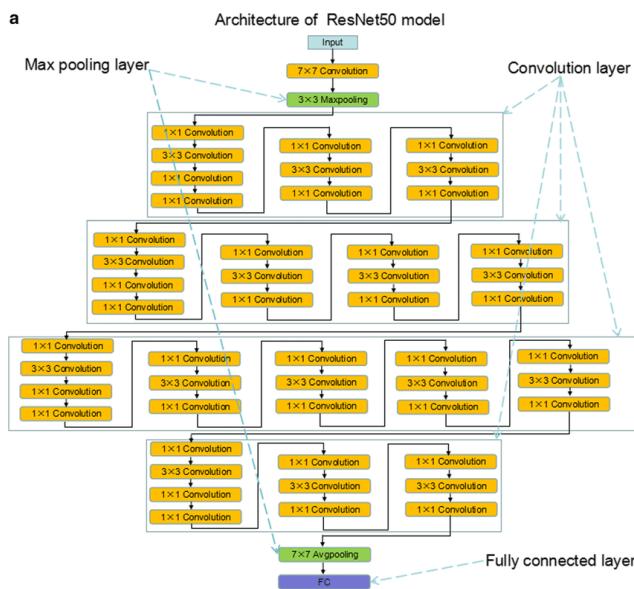
Where x is the input to the block, $F(x)$ is the output from the network layers within the block, $H(x)$ is the final output of the block. This approach helps to avoid the issue of disappearing gradients, allowing for deeper networks by guaranteeing that the gradient may propagate back through the layers without significant loss of information.

Figure 64 provides an in-depth view of the internal configuration of the ResNet50 model which has an initial layer consisting of a 7×7 convolution by 3×3 max pooling. The following blocks are made up of

repeated sets of 1×1 , 3×3 , and 1×1 convolutions, with an objective of reducing the dimensionality, learning spatial structures, and restoring dimensions. This process concludes with a global average pooling layer and an entire linked layer as output.

Figure 64

Detailed Architecture of the ResNet50 Model



Note. This figure is taken from the paper by (Mandingo et al., 2023)

Table 21

ResNet Model pseudocode

Step	Pseudocode
Load Pre-trained Model	model = ResNet50(weights='imagenet', include_top=False)
Image Input Processing	x = Resize and reshape input image to 224x224x3
Apply ResNet50	x = model(x)
Metadata Input Processing	m = input_meta

Step	Pseudocode
Process Metadata	$m = \text{Dense layers processing on metadata}$
Concatenate Features	$\text{combined} = \text{concatenate}([x, m])$
Output Layer	$\text{output} = \text{Dense}(\text{num_classes}, \text{activation}=\text{'softmax'})(\text{combined})$
	$\text{model.compile(optimizer='adam',}$
Compile Model	$\text{loss}=\text{'sparse_categorical_crossentropy', metrics=['accuracy']}$
Train Model	$\text{model.fit}([\text{image_data}, \text{meta_data}], \text{labels}, \text{epochs}=30,$ $\text{batch_size}=32)$

Note. The pseudocode has been adapted from the work by He et al. (2016)

Table 22

Formulae for each component of ResNet50 model along with their corresponding equations

Component	Equation
Initial Convolution	$Y=W \cdot X + b$
Batch Normalization	$Y=\gamma(X-\mu/\sigma)+\beta$
ReLU Activation	$Y=\max(0,X)$
Convolution (in blocks)	$Y=W \cdot X + b$
Shortcut Connection	$Y = X_{\text{input}} + X_{\text{conv}}$
Global Average Pooling	$Y=\text{avg}(X)$

Component	Equation
Metadata Processing	$Y = W \cdot M + b$
Concatenation	$Z = \text{concat}(X, Y)$

Model Optimization. Multiple strategies were employed to optimize the ResNet50 model, integrating it with metadata to improve accuracy and performance. To start with, altering the learning rate through dynamic scheduling or decaying can significantly improve the models speed of convergence and ultimate accuracy(He, Zhang, Ren, & Sun, 2016). Furthermore, using L2 regularization with a well calibrated regularization factor helps to reduce overfitting which is a common issue in models like ResNet50. Another useful strategy is to change the batch size, since different sizes can affect how the network learns, with the smaller batches typically resulting in enhanced generalization.

Dropping Out layer with maximum dropout rate can also help to prevent overfitting through implementing a type of model averaging that improves the network adaptability. And a further important technique is to stop early stopping, which pauses the training when validation performance no longer gonna increase, reducing overfitting and saving the computing resources. In addition to this more stronger data augmentation approaches can improve the ability of models to generalize to new, previously unknown data. Finally, optimizing the fusion of image and metadata features could unlock new levels of performance, especially by exploring various methods of combining these distinct data types (He et al., 2016). These optimization techniques not only goal for improvising accuracy and efficiency of the model but also enhance its applicability to real world scenarios by ensuring it operates effectively on new, unseen data.

Model Supports

The accurate classification of skin lesion images was achieved through a robust setup featuring an Intel Core i7 processor, NVIDIA RTX 30 Series GPU, and 16GB RAM. These hardware components provided the necessary computational power, GPU acceleration, and memory capacity for efficient processing of large datasets and complex deep learning models. The operating system requirements options include Windows 11 for a familiar environment or macOS for systems with Apple's M1/M2 chips or compatible hardware, ensuring optimized performance for deep learning tasks. GoogleColab Pro was used for the programming tasks since it seamlessly integrates with Google Cloud Platform where the data for the project has been stored. Python was the programming language used throughout the project.

The deep learning models implemented for this study employed a wide range of tools and libraries of Python for data extraction, data preprocessing, data engineering, data preparation and finally for modeling and measuring performance.

For preprocessing the data, libraries like Pandas, and its methods like read_csv, isnull, dropna, isduplicated were used for efficient handling of the data. The numeric features were scaled using MinMaxScaler class of Scikit-learn library and all the categorical features were encoded using get_dummies method of Pandas library. And for reading the images Image method of PIL Library was used and for cleaning the images of hair, Dull Razor Method (Gururaj et al., 2023) was used which included various methods of OpenCV library like 'imdecode', 'cvtColor', 'morphologyEx', 'GaussianBlur', 'threshold' and 'inpaint' were used to remove hairs from the images. From Scikit-learn library StandardScaler and PCA (El-Gamal, 2013) classes were used to normalize the pixel values and for dimensionality reduction, respectively.

For data preparation, which involves splitting the data for training, validating, and testing purposes, train_test_split method from Scikit-learn library was used to split both the metadata and the

image data into the train, test, and validation sets. For augmenting data `ImageDataGenerator` class from Keras was used. For modeling, Keras and TensorFlow libraries were used. Methods such as `fit`, `predict` and `evaluate` were used to train, validate, and test the model's performance. Finally for evaluating the model various metrics such as `f1_score`, `precision_score`, `recall_score`, `confusion_matrix`, and `accuracy` from scikit-learn library were used to provide a comprehensive assessment of the model.

Additionally, visualization libraries such as Matplotlib and Seaborn were employed to visualize and have a better understanding of the data. Apart from this '`matplotlib.pyplot`' was used to visualize training and validation losses and heatmap of seaborn was used to visualize the confusion matrix, which gives an overall understanding of the performance of the model.

Table 23 provides a brief summary of all the libraries and methods used in modeling.

Table 23

Required Libraries, Methods, Usage of these Methods

	Library	Method	Usage
Pandas	DataFrame	<code>read_csv</code> , <code>shape</code> , <code>head</code> , <code>get_dummies</code>	Data loading, understanding, preprocessing and encoding categorical features.
NumPy	Numeric Calculations	<code>seed</code> , <code>argmax</code> , <code>reshape</code>	For reshaping images, maintenance of consistency in random splitting
PIL	Handles images	<code>Image</code>	To read the images
OpenCV	To transform images	<code>imdecode</code> , <code>cvtColor</code> , <code>morphologyEx</code> , <code>GaussianBlur</code> , <code>threshold</code> and <code>inpaint</code>	To clean images of hair
matplotlib, pyplot seaborn		<code>pyplot</code> , <code>heatmap</code>	For plotting confusion matrix, and for visualization of data analysis

	Library	Usage	Library
Keras	keras.layers keras.models keras.optimizers	Conv2D, Flatten, Dropout, Dense, Input, concatenate, AveragePooling2D. model Adam	For model architecture, optimizer
Scikit-learn	sklearn.decomposition sklearn.preprocessing sklearn.model_selection sklearn.metrics	PCA StandarScaler, MinMaxScaler train_test_split confusion_matrix, f1_score, precision_score, recall_score	For dimensionality reduction, normalizing the features and pixel values, for splitting the data and for evaluating the model.

Model Architecture and Dataflow

The HAM10000 dataset (Tschandl et al., 2018) underwent initial data preprocessing and data transformations to prepare it for the modeling phase. Under preprocessing, missing age values were imputed, categorical columns: sex and localization were encoded, and hair in images were removed using the dull razor method. Under Data transformation age feature and pixels of lesion images were normalized to range [0,1] for faster convergence. To reduce the computational load and to reduce dimensionality, Principal Component Analysis (PCA) (El-Gamal, 2013) was performed on the images. To address the class imbalance in the dataset, especially for less represented classes like ‘df’ and ‘vasc’, data augmentation was performed using Keras ImageDataGenerator, applying random flipping, zooming and rotation to images (Syafa’ah et al., 2021). The preprocessed data, both image data and metadata, was then divided into training, testing, and validation sets in the ratio of 60:20:20, which is crucial for evaluating model’s performance on unseen data, tuning hyperparameters, and assessing model’s generalization ability.

The prepared data is initially fed to a Convolutional Neural Network (CNN). This baseline model has several layers, including input, convolutional, pooling, activation, flattening, fully connected, and output layers. The convolutional layers extract features using filters. The pooling layers help in dimensionality reduction. ReLU activation functions introduce non-linearity, and dropout layers are used to prevent overfitting. This model was used as a base to compare with more advanced models.

Following the CNN, the Convolutional Neural Network-One-Versus-All (CNN-OVA) model was implemented. This model uses CNN in conjunction with OVA, where for N classes, N binary classifiers are first trained and then merged together. Since each binary classifier determines whether an instance belongs to a class or not, applying CNN in this model using the One-Versus-All procedure offers vast improvements in performance due to the multi-class dataset.

The DenseNet121 model was employed. DenseNet121 features dense blocks with multiple convolutional layers where each layer receives inputs from all its previous layers, for efficient feature propagation and reuse. Transition layers were included to reduce the number of feature maps and spatial dimensions. Dropout layers and L2 regularization were applied to avoid overfitting.

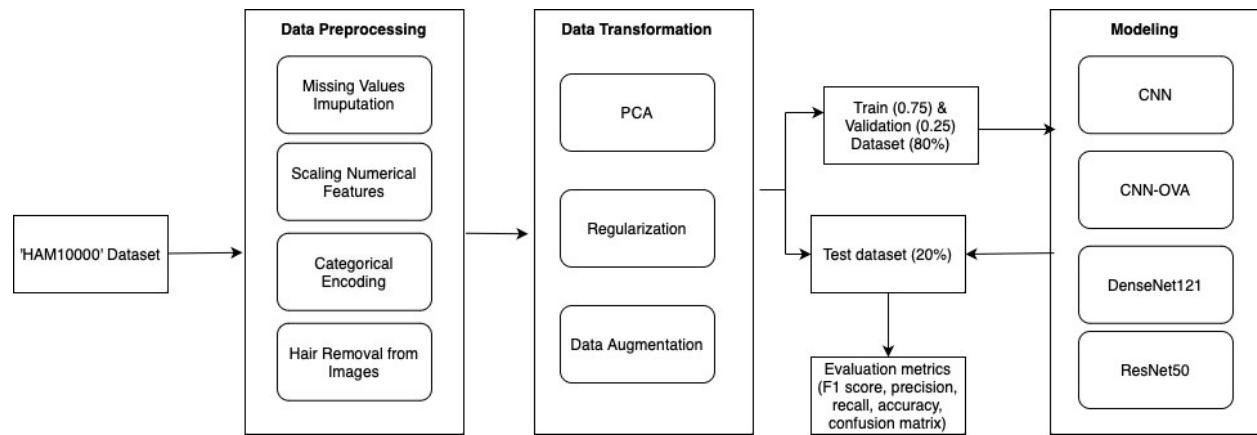
Lastly, the ResNet50 model was implemented, utilizing its innovative residual learning framework to address the challenges of training deeper networks. This model's deep structure, combined with its ability to maintain performance across numerous layers, proved particularly effective for the challenging task of skin cancer classification, achieving high accuracy and robust performance.

After training, the model is then evaluated against the test dataset, which is reshaped into 3D format of (15,10,1). The model is evaluated against a set of evaluation metrics: F1-score, recall, precision, and accuracy. Confusion matrix is also plotted to get clear insights into the predictions of the model.

The dataflow of the project is illustrated in the Figure 65.

Figure 65

Dataflow through the Project



Model Comparison and Justification

Deep learning models like CNN, CNN-OVA, DenseNet121, and ResNet50 were implemented in the training of the preprocessed and prepared data of the project to achieve proper classification. The testing part of the project was executed on four models. The result from all four models was similar because of the very small size of the dataset and also because PCA is implemented on the images, it gave a higher accuracy. As model architecture says, DenseNet121 is a densely connected convolutional network that improves gradient flow and promotes feature reuse with dense connections. ResNet50 is an architecture model based on the residual learning framework. This helps train the deeper networks by the vanishing gradient problem. CNN-OVA (One-Versus-All) is a strategy in which multiple binary classifiers are trained for distinguishing one class versus the rest. CNN is a simpler version of a convolutional neural network that does not have the other strategies implemented, like dense connections or residual learning.

All the four models are basically designed for image data, but they can be twisted into other forms of data, such as tabular or text, with proper preprocessing. DenseNet121 and ResNet50, based on their strong architecture and regularization techniques, are expected to perform well on both small and

large datasets. The CNN models may function less well with a smaller dataset, where the risk of overfitting increases, with the plain CNN model without advanced strategies. Overfitting is carried out with all models. DenseNet121 exploits both dense connections and dropout, and the residual connections used by ResNet50 are good at this. Even though the CNN-OVA approach is overfitting-robust due to the problem decomposition in simpler binary classifications, the plain CNN is sensitive to overfitting without careful regularization.

DenseNet121 and ResNet50 are much more complex, with a higher level of architecture, higher preprocessing, higher training time, and huge computational resources (GPU). The other added layer of complexity in the CNN-OVA is with the multiple binary classifiers, and the plain CNN becomes comparably simple in terms of complexity. DenseNet121, being effective in reusing features, improving gradient flow, and bringing great speed of convergence, is very useful for complex tasks like image classification. But, it is highly complex in nature and requires huge computation. With the residual connection, ResNet50 can have the power to train very deep networks and not suffer from the vanishing gradient. But it may become hard to train and there are chances of overfitting without the right regularization. Multiclass classification becomes quite easy with CNN-OVA since classification is reduced to many binary problems. However, the strategy comes along with an increase in complexity and training time, as multiple classifiers are used. This structure of CNN model is not complex, and it has been effective for a lot of image classification tasks; at the same time, it suffers very high risks of overfitting and needs more tuning for good performance. Table 24 summarized the comparison of each of the four models.

Table 24*Comparison of Models*

Characteristic	DenseNet121	ResNet50	CNN-OVA	CNN
Architecture	Parallel processing through dense connections, non-linear model	Parallel processing with residual connections, non-linear model	Multiple linear models combined for multi-class classification	Linear model with sequential convolutional layers followed by fully connected layers
Data Types	Image, Numerical and Categorical data with preprocessing	Image, Numerical and Categorical data with preprocessing	Image, Numerical and Categorical data with preprocessing	Image, Numerical and Categorical data with preprocessing
Impact of Small Dataset	Performs well with regularization	Performs well with improved residual connections	Handles small data by breaking down into simpler classifications	Tends to overfit, required proper regularization
Overfitting Issues	Moderate	Moderate	Moderate	High
Preprocessing Extent Required	High preprocessing, significant memory and computational requirements	High preprocessing, significant memory and computational requirements	High complexity due to multiple classifiers	High preprocessing, simpler structure but high parameter count
Training Time	Long due to complex architecture	Long due to complex architecture	Long due to multiple binary classifiers	Moderate
Space Complexity	High	High	High	High
Computational Complexity	High	High	High	High

Model Evaluation Methods

The skin lesion images are classified into one of the seven classes using the deep learning models. However, to make the model reliable in the real-world, it is crucial to evaluate the performance of the models using essential evaluation metrics. These metrics also help in finding out the best

performing model out of the four different models upon which the dataset is trained. Accuracy, Precision, Recall, F1 Score are the metrics utilized to understand the performance of the models on the unseen data, eventually leading to the selection of a reliable model. Apart from these, to clearly distinguish between the actual and predicted labels, a Confusion matrix was employed.

Accuracy

Accuracy is an important metric for Skin Cancer Lesion Classification, which measures the proportion of correctly classified instances, combining both true positives and true negatives against the total number of instances (Yang et al., 2023). A high accuracy indicates that model is classifying a greater share of instances correctly, while a low accuracy means that the model is making a lot of mistakes in classifying. However, in cases of medical imaging where datasets can be heavily imbalanced like skin cancer lesions where non-cancerous instances far outnumber the cancerous ones, accuracy can be quite deceiving. As the model performs good at predicting the majority class but fail in predicting the minority class which is more important class of cancerous lesions. Yang et al. (2023) gives the formula for computing accuracy given in Equation 4

$$\text{Accuracy} = \frac{\text{Total True Predictions}}{\text{Total Number of Predictions}} \quad (4)$$

Precision

Vujovic (2021) describes precision as a metric in classification performance; it measures the proportion of correct positive predictions. It can be computed as the ratio of the number of truly predicted positive classes (TP) to the total number of predicted positive classes—true positive and false positive (TP + FP). Precision tells how many of the cases predicted as positive are correct, therefore ranging from 0 to 1. In other words, an ideal precision value of 1.0 would mean that all positive predictions are indeed positive, and a low value of 0.0 would mean none of the positive predictions is correct (Vujovic, 2021).

$$\text{Precision} = \frac{\text{Total True Positives}}{\text{Total Number of Positives}} \quad (5)$$

Recall

Recall, or sensitivity, is calculated by the actual positive instances that the model correctly classifies, divided by the total number of instances that are true positive and false negatives. It measures how well the model identifies positive cases and ranges in value from 0 to 1. A recall of 1.0 means all positive instances are identified correctly by the model, and a recall of 0.0 means none is identified (Hossin & Sulaiman, 2015).

$$\text{Recall} = \frac{\text{Total True Positives}}{\text{True Positives} + \text{True Negatives}} \quad (6)$$

F1-Score

F1-score is given by the harmonic mean of precision and recall, providing a balance between both these metrics. Chao et al. (2022) proposed F1-score for evaluating the performance of skin cancer lesion classifiers as they both are contradictory measures i.e., when precision is high the recall is low and inversely when recall is high, the precision will be low. So, in order to comprehensively consider both these indicators F1-score is evaluated. Yang et al. (2023) gives the formula of F1-score, a harmonic mean of precision and recall, which is given in Equation 7.

$$\text{F1 Score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (7)$$

Confusion Matrix

Grandini et al. (2020) describes a confusion matrix as a 2×2 table recording the frequency of concurrence between two raters—true (actual) classification and predicted classification. The columns have been used to show what the model has predicted, whereas the rows are used to indicate the true classifications. The classes are listed in the same order in both rows and columns, such that the truly classified instances are on the main diagonal from top left to bottom right and represent the number of

times the two raters agree. An example of a confusion matrix is depicted in Figure 66, for general understanding.

Figure 66

Example of Confusion matrix

		PREDICTED classification				Total
Classes		a	b	c	d	
ACTUAL classification	a	6	0	1	2	9
	b	3	9	1	1	14
	c	1	0	10	2	13
	d	1	2	1	12	16
	Total	11	11	13	17	52

Note. This is image it taken from Grandini et al. (2020)

ROC and AUC Curves

The developed model with DenseNet121 is intended for multiclass classification to classify images of skin cancer into one of the seven classes. Thus, ROC and AUC would not be practical. They are designed for binary classification, and it is hard to interpret these scores in the case of multiclass problems. The ROC curve shows the relationship between true positive rates and false positive rates, and in the case of binary classifiers, it is all right. This is because multiclass classification can have many classes, with each class being considered a "positive," hence the rest being "negatives." In this case, there will be a number of ROC curves, whereas their corresponding AUC values are hard to aggregate and interpret from the ROC results. Therefore, the suitable metrics for the proper evaluation of a multiclass model are one-vs.-rest ROC curves and precision-recall curves.

Model Validation and Evaluation Results

The performance of the implemented models in predicting skin cancer type based on lesion images and metadata, and the effect of Principal Component Analysis (PCA) and L2 Regularization on the predictions using CNN model were evaluated on the test dataset against evaluation metrics accuracy,

precision, recall, and F1 score to get a clear understanding of the performance. Apart from these metrics scores, confusion matrices for each of these models are visualized, for a better understanding and to paint a clear picture of the misclassifications made by the models.

Convolutional Neural Networks

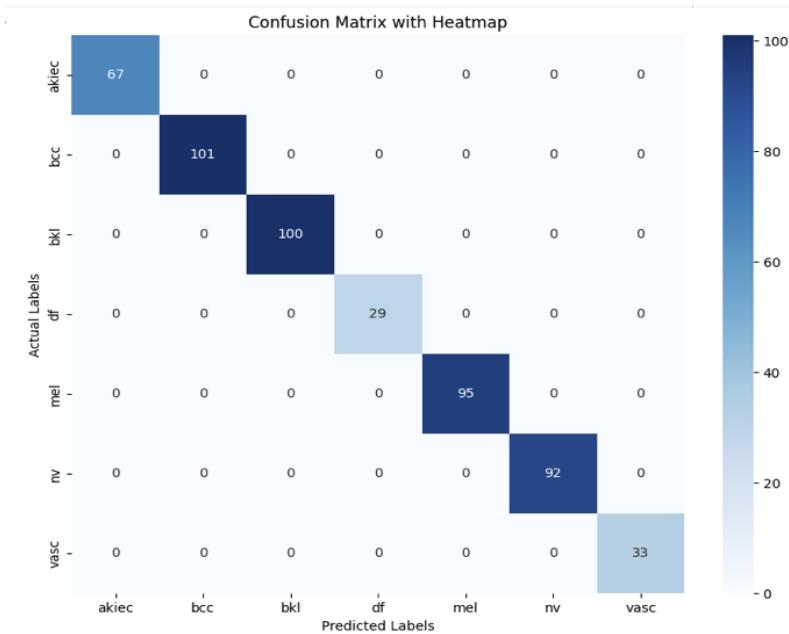
Initially, the first CNN model, where the input is the raw data without any data transformations i.e., without PCA or L2 Regularization, achieved an accuracy of 100% and all the other metrics, including precision, recall and F1 score are 1.00. The confusion matrix for this model, which is shown in Figure 67, shows no misclassifications at all. These scores clearly indicate that the model is heavily overfitted and it has memorized all the training dataset instead of learning and generalizing.

The second CNN model where PCA was applied on images for dimensionality reduction, achieved an accuracy of 99% and all the other metrics scores, including precision, recall and F1 score are 0.99. The confusion matrix for this model, shown in Figure 68 presents few misclassifications made by the model, i.e., misclassifying one class as another, for example the model misclassified 'df' class instance as 'mel' class. Though the introduction of PCA in the basic CNN model helped a bit in preventing overfitting, it is still prevalent as all the metrics scores are relatively high at around 99%.

Finally, in the third CNN model L2 Regularization was introduced along with PCA to reduce overfitting which persisted even after applying PCA to the model. This model penalizes the larger weights by introducing a penalty term to the loss function during training which helps in preventing overfitting the model. This algorithm achieved an accuracy of 94.78%, precision of 0.95, recall of 0.94 and F1 score of 0.96 indicating that the model is performing better on new unseen data. The confusion matrix for this model, shown in Figure 69 presents more misclassifications when compared to other two model variants, especially in 'df' and 'vasc' classes. This clearly indicates that the CNN model with PCA and L2 Regularization is generalizing well to new unseen data.

Figure 67

Confusion Matrix of baseline CNN model without PCA and L2 Regularization

**Figure 68**

Confusion Matrix of CNN model with PCA

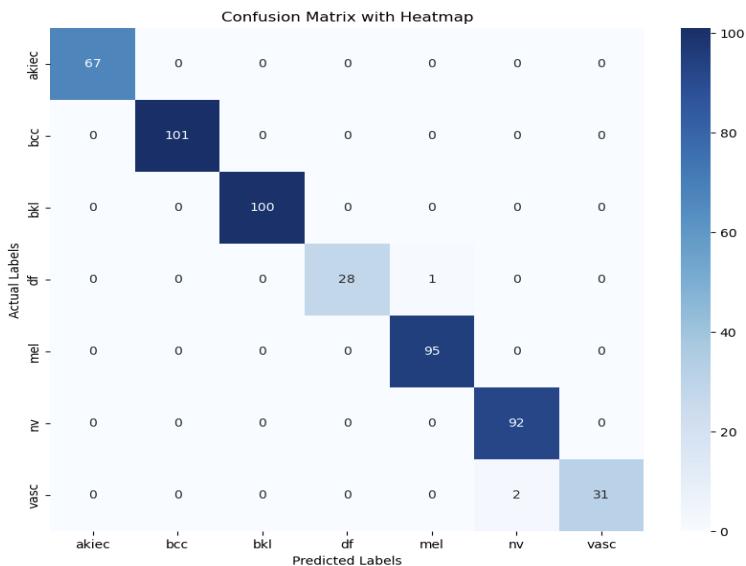


Figure 69

Confusion Matrix of CNN model with PCA and L2 Regularization

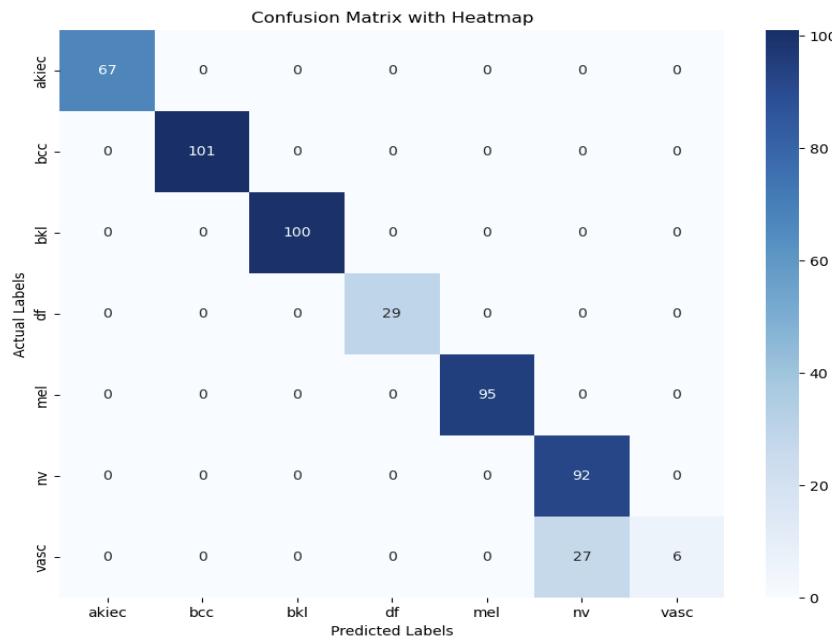


Table 25 shows the evaluation metrics compared across the three models discussed above.

Table 25

Comparison of Model Performances

Evaluation	Model 1 (CNN without PCA and Regularization)	Model 2 (CNN with PCA)	Model 3 (CNN with PCA and L2 Regularization)
Metric			
Accuracy	1.00	0.99	0.94
F1 Score	1.00	0.99	0.96
Precision	1.00	0.99	0.95
Recall	1.00	0.99	0.94

It is observed that the precision, recall and F1 score of CNN model with PCA and L2 regularization are slightly lower when compared with CNN model with PCA alone, indicating that the

model is making more errors than the latter model but is still detecting most of the actual positive cases, which is a good trade-off for overfitting and generalization of the model towards new unseen data.

To conclude, though the basic CNN model achieved perfect scores and accuracy, it is not reliable in real-time scenarios as it is heavily overfitted and is not generalizing well on new unseen data. And while the second model, CNN with PCA is generalizing well, it is still clearly overfitted and is struggling with some classes and finally CNN model with both PCA and L2 Regularization employed performs better by generalizing well on new unseen data making it most suitable for skin cancer lesions classification.

Further, the best model which is CNN with PCA and L2 Regularization is selected to perform hyperparameter tuning to arrive at the best performing CNN based classifier.

Hyperparameter Tuning. Hyperparameter tuning is essential in optimizing model's performance. It includes adjusting parameters of the model, such as learning rate, batch size, number of hidden layers, number of neurons per layer and in this case the regularization parameter of L2 Regularization can also be tuned for optimizing the performance of the model. This optimization technique helps improve the performance of the model and finds a balance between the accuracy and the convergence speed of the model, and also helps in preventing overfitting and underfitting.

This study experimented with three variants of the same model, CNN with PCA and L2 Regularization, by tweaking the regularization parameter while keeping all other parameters the same. The regularization parameters chosen for this purpose are 0.01, 0.02 and 0.05. Table 26 summarizes the metrics achieved by the model with each of these values.

Table 26

Comparison of CNN model with PCA and L2 Regularization with different regularization parameters

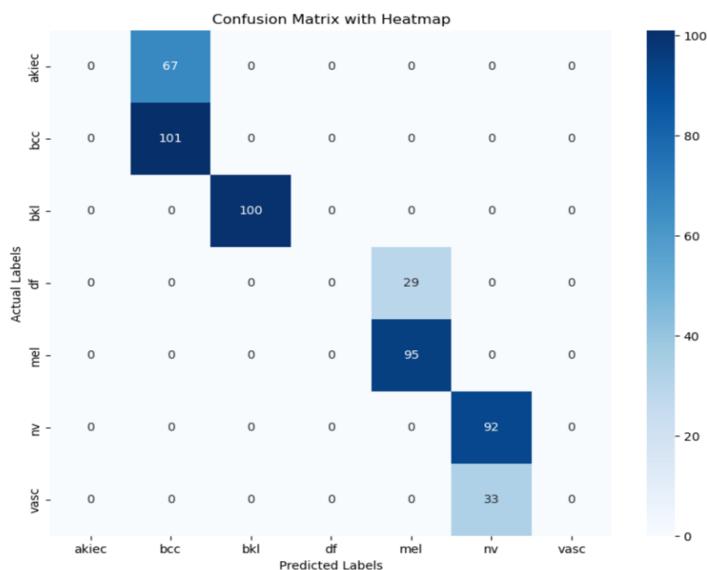
Regularization parameter	Accuracy	F1 Score	Precision	Recall
0.05	0.75	0.85	0.58	0.75
0.02	0.79	0.86	0.66	0.79

Regularization parameter	Accuracy	F1 Score	Precision	Recall
0.01	0.94	0.96	0.95	0.94

CNN model with PCA and L2 Regularization with 0.05 regularization parameter. For a regularization parameter of 0.05, the model did not perform well when compared with the other two model variants. It has the least accuracy, precision, recall and F1 score among the three. This indicates that the model does not converge efficiently and that there are many misclassifications which can be observed in the confusion matrix given in Figure 70.

Figure 70

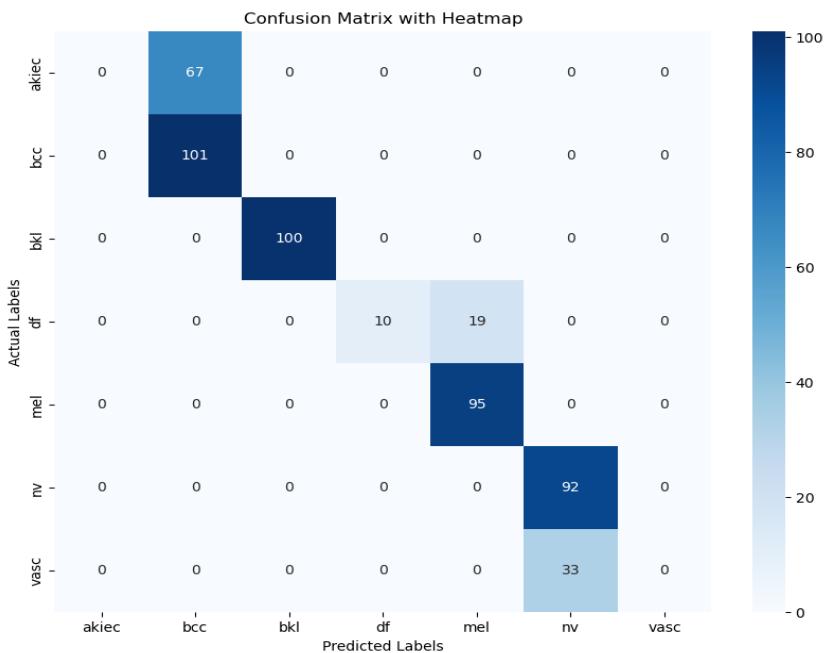
Confusion matrix of model with 0.05



CNN model with PCA and L2 Regularization with 0.02 regularization parameter. With the regularization parameter of 0.02, the model achieved a moderate performance of about 79%. Though there is a slight improvement over the previous model, there are still many misclassifications which can be observed in the confusion matrix given in Figure 71.

Figure 71

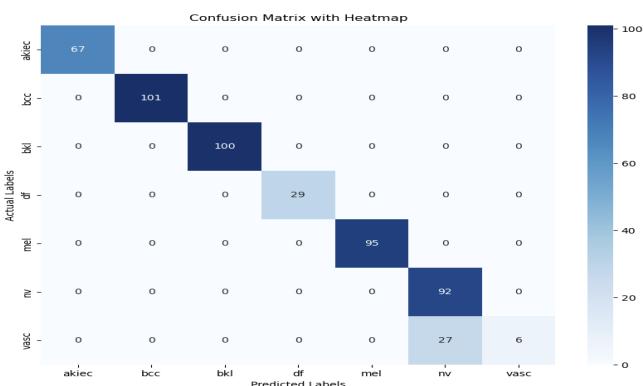
Confusion matrix of model with 0.02



CNN model with PCA and L2 Regularization with 0.01 regularization parameter. Among the three, the model with 0.01 regularization parameter shows best accuracy, precision, recall and F1 score. This indicates that the model is converging well, learning well in classifying data. The confusion matrix of this model in Figure 72 shows that though it is misclassifying some of the classes, it is accurately distinguishing between most of the classes.

Figure 72

Confusion matrix of model with 0.01



To conclude, model with 0.01 regularization parameter is performing well leading to a well balanced stable model, with minimum number of misclassifications. The higher recall and precision indicate that the model is performing well in identifying both true positive and true negative cases. Whereas the other two models have a high recall value compare to precision, indicating that these models are good at identifying True positives but also have a higher rate of False Positives, which could lead to unnecessary panic among the patients and unnecessary further testing or treatment.

CNN-OVA

Implementation of CNN-OVA (Convolutional Neural Network – One versus All) model in skin cancer classification project has given great results. Measured the model's performance using metrices like precision, recall, and confusion matrix. The model showed test loss of 0.869, with good accuracies in classifying skin lesions. The model achieved 100% accuracy in classifying classes zero and one images. However, it demonstrated lower recall and precision for classes four and five.

As it was showing 100% accuracy for few classes and not performing that great in some classes led to the implementation of regularization. The main aim of regularization is to overcome overfitting and increase the capability of model to perform well in the unseen data situations. After regularization, the precision and recall were enhanced for the classes which are underperformed previously. The recall for class three improved from 0.5862 to 0.7609 and precision for class four increased from 0.7778 to 0.8689. Additionally, the accuracies of the models which showed 100% also decreased. Table 27 provides the comparison of the CNN-OVA model parameters before regularization and after regularization.

Table 27*CNN-OVA Before and After Regularization*

Parameters	Before Regularization	After Regularization
Number of Conv2D Layers	2	2
Filters in First Conv2D	32	32
Filters in Second Conv2D	64	64
Kernel Regularization	None	L2(0.01)
Activation Function	ReLU	ReLU
Dropout Rate	0.0	0.5
Size of Dense Layer	512	512
Batch Size	32	32
Epochs	30	30
Optimizer	Adam	Adam
Learning Rate	0.001	0.001
Loss Function	Binary Crossentropy	Binary Crossentropy

Table 28 compares the metrices of the models before regularization and after regularization.

Table 28

Comparison of the metrices of the models before regularization and after regularization

Metric	Before Regularization	After Regularization
Test Accuracy for Class 0	1.0000	0.9768
Test Accuracy for Class 1	1.0000	0.8801
Test Accuracy for Class 2	0.9961	0.9652
Test Accuracy for Class 3	0.9768	0.9516
Test Accuracy for Class 4	0.9323	0.8820
Test Accuracy for Class 5	0.9188	0.9188
Test Accuracy for Class 6	0.9594	0.9710
Confusion Matrix for Class 0	[[450 0] [0 67]]	[[448 2] [10 57]]
Confusion Matrix for Class 1	[[416 0] [0 101]]	[[397 19] [43 58]]
Confusion Matrix for Class 2	[[416 1] [1 99]]	[[412 5] [13 87]]
Confusion Matrix for Class 3	[[488 0] [12 17]]	[[486 2] [23 6]]
Confusion Matrix for Class 4	[[398 24] [11 84]]	[[392 30] [31 64]]
Confusion Matrix for Class 5	[[404 21] [21 71]]	[[403 22] [20 72]]
Confusion Matrix for Class 6	[[482 2] [19 14]]	[[483 1] [14 19]]

Metric	Before Regularization	After Regularization
Precision for Class 0	1.0000	0.9661
Precision for Class 1	1.0000	0.7532
Precision for Class 2	0.9900	0.9457
Precision for Class 3	1.0000	0.7500
Precision for Class 4	0.7778	0.6809
Precision for Class 5	0.7717	0.7660
Precision for Class 6	0.8750	0.9500
Recall for Class 0	1.0000	0.8507
Recall for Class 1	1.0000	0.5743
Recall for Class 2	0.9900	0.8700
Recall for Class 3	0.5862	0.2069
Recall for Class 4	0.8842	0.6737
Recall for Class 5	0.7717	0.7826
Recall for Class 6	0.4242	0.5758

As per the comparison, the model is working very well by accurately classifying the images.

Model is overfitted for classes zero and one, but after regularization model has 0.9768 accuracy for class zero and 0.8801 for class one. This indicates the model's classifying capability was increased and model

became more general as this prepares the model to perform well on unseen data. Table 5 compares the metrics of CNN-OVA before and after regularization.

Table 29

Metrics Comparison of Different Models Before and After Regularization

Model	Condition	Training Accuracy	Validation Accuracy	Test Accuracy	F1 Score	Precision	Recall
CNN - OVA	Before Regularization	98.83% (Avg)	97.68% (Avg)	97.72% (Avg)	0.9786 (Avg)	0.9231 (Avg)	0.9045 (Avg)
	After Regularization	99.35% (Avg)	93.59% (Avg)	93.63% (Avg)	0.7892 (Avg)	0.8103 (Avg)	0.6182 (Avg)

DenseNet121

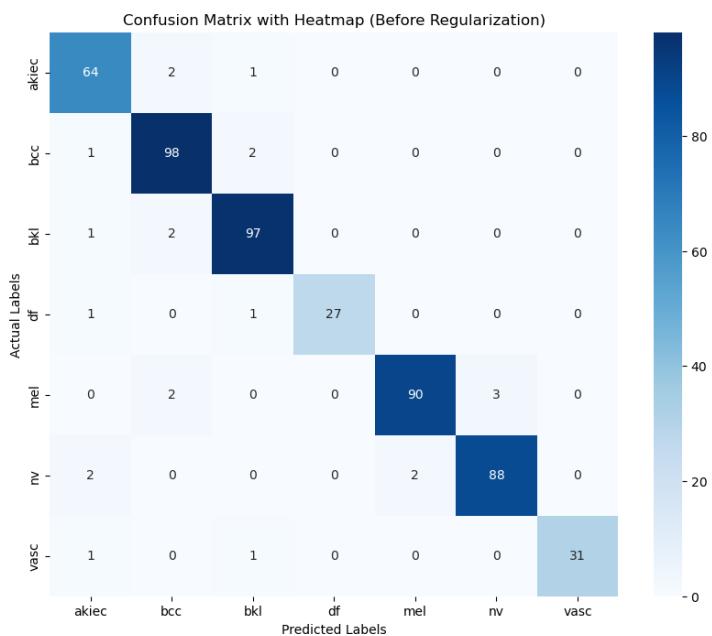
For evaluating the DenseNet121 model, several metrics were used, including accuracy, precision, recall, and F1 score. To support these results, a confusion matrix was generated that visually represents the true positive, true negative, false positive, and false negative counts for each class. ROC and AUC curves were not utilized due to their limitations in multiclass classification, as previously explained.

Evaluation Results of Base Model. Images from the ‘Ham10000’ were passed through the input layers along with the metadata, followed by the DenseNet121 architecture. The trained model provided results, 98.97% training accuracy and 99.42% validation accuracy, and 95.02% test accuracy. The model achieved a precision, recall, and F1 score of 0.9420, 0.9434, and 0.9467, respectively. The obtained results have supported the inference that the model performed well not only on the training set but also on the validation set, meaning that underlying patterns in the data were learned effectively. However, test accuracy and metrics reveal some inferior scores, showing a hint of model overfit, meaning the

model learned well from the training data but did not generalize well in the new data. Further analysis of the confusion matrix as depicted in Figure 73, indicated that even though it illustrated good performance in most of the classes, it singled out the classes for which the model had classification problems, which resulted in a misclassification.

Figure 73

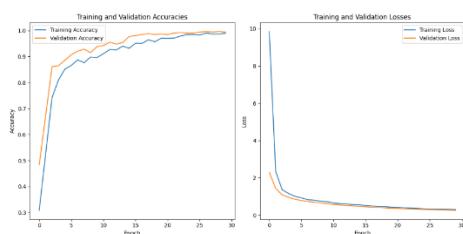
Confusion Matrix for DenseNet121 Before Regularization



The training and validation accuracies as shown in the Figure 74, being high and robust in other metrics, respectively, show that the model can manage the complexity of the task of classifying skin cancer; meanwhile, the overfit condition requires the application of generalizing techniques.

Figure 74

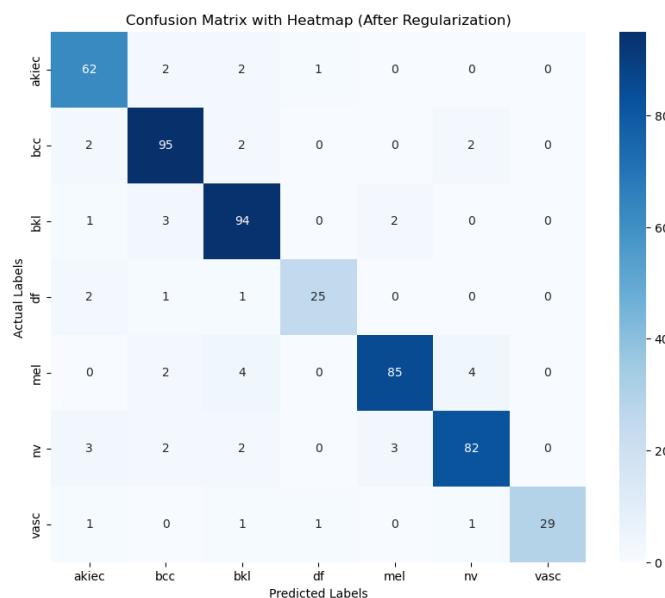
Train and Validation Accuracy & Losses Curves of the Base Model



Evaluation Results After L2 Regularization and Dropout Introduction. The DenseNet121 model was re-trained with addition of L2 regularization and dropout layers to avoid overfitting. The L2 regularization coefficient is taken as 0.001. Dropout layers are added to the image branches of the model. This model obtained a training accuracy of 98.97%, validation accuracy of 99.42%, and test accuracy of 93.30%. The model delivered a precision of 0.9123, recall of 0.9035, and F1 of 0.9312. Thus, the test accuracy is a little less, while all the other measures are related to the non-regularized model, which has proved the regularization techniques work well and decrease overfitting for the model to better generalize for unseen data. The confusion matrix as depicted in Figure 75, was more balanced for the classes. It showed many lower extremity misclassifications. The final result is more consistent across performances.

Figure 75

Confusion Matrix for DenseNet121 after Regularization

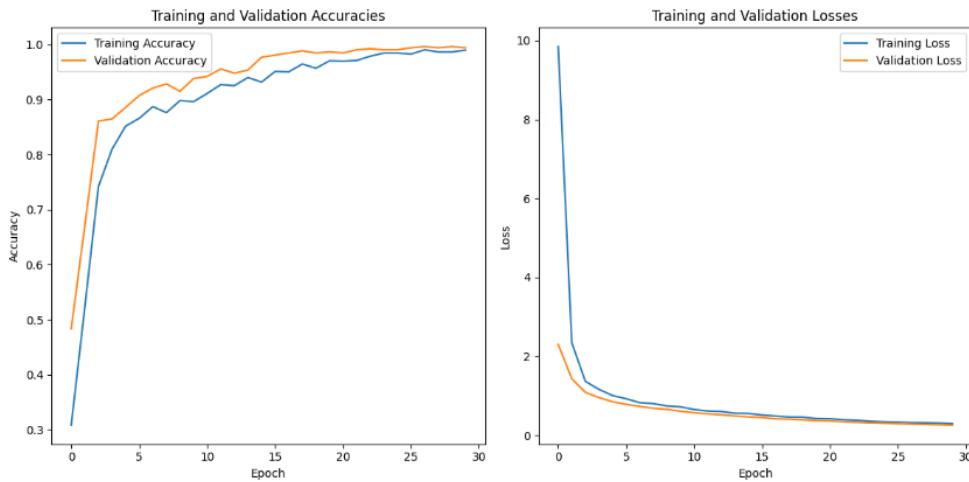


The training-validation accuracy, and loss plots as illustrated in Figure 76, further provided evidence of regularization. A much better trade-off between the precision-recall curves and the learning

process of the loss plots with much more stable learning, in comparison with the overfitting, was observed.

Figure 76

Train and Validation Accuracy & Losses Curves of the Regularized Model



Comparison of variations of the DenseNet121 model, which are base DenseNet121 model and Model with regularization implemented, is summarized in Table 30.

Table 30

Comparison of Model variations through Evaluation Metrics

Model	Condition	Training	Validation	Test	F1 Score	Precision
		Accuracy	Accuracy	Accuracy		
DenseNet121	Before Regularization	98.97%	99.42%	95.02%	0.9467	0.9420
DenseNet121	After Regularization	98.97%	99.42%	93.30%	0.9312	0.9123

These comparisons clearly depict the importance of regularization in enhancing the generalizability of the model.

ResNet50

To evaluate the ResNet50 model, metrics such as accuracy, precision, recall, and F1 score were employed. Additionally, a confusion matrix was created to visually display the true positive, true negative, false positive, and false negative counts for each class. ROC and AUC curves were not used because of their limitations in handling multiclass classification, as previously discussed.

Evaluation Results of Base Model. In the case of evaluating the model the data is split into 80% training data and 20% testing data. Further training data is divided into 75% training and 25% validation dataset, and the model was created. The model utilizes the TensorFlow and the ResNet50 architecture to process the dermoscopic image data alongside metadata for classification.

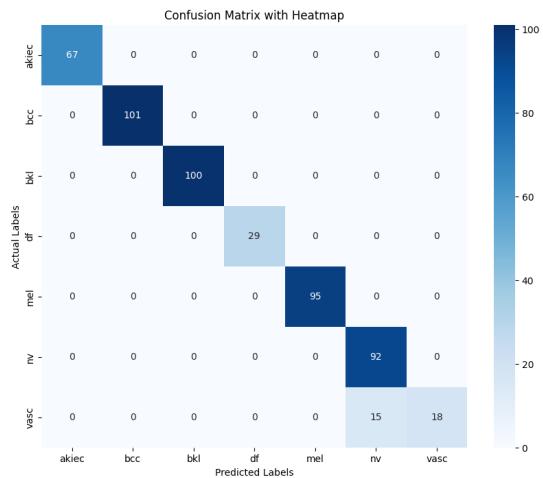
For baseline models using ResNet50, typically involved the configurations architecture minimally, by leveraging pre-trained weights. This basically involves using the ResNet50 model as the main feature extractor without its original classification top layer. This setup ensures that the complex features that are learned by the ResNet50 on the ImageNet are utilized. And then the model employs the Adam optimizer with the learning rate of 0.001 and uses the sparse categorical cross-entropy for the loss function.

The first metric which is derived from the model is accuracy in which it demonstrated well and gave strong performance with baseline model by achieving a final training accuracy of 98.97% and for the higher validation with 99.42%. However, it maintained a high accuracy of 97.10% on the test set, in which it indicates the better performance. And the model also scored well in precision, recall and f1 score.

The confusion matrix displayed in below Figure 77 shows how well the classification model predicts across different skin lesions, key observation is akiec," "bcc," and "bkl" classes are accurately predicted with no errors but "mel" and "nv" depicts the good prediction accuracy but have some misclassifications between them.

Figure 77

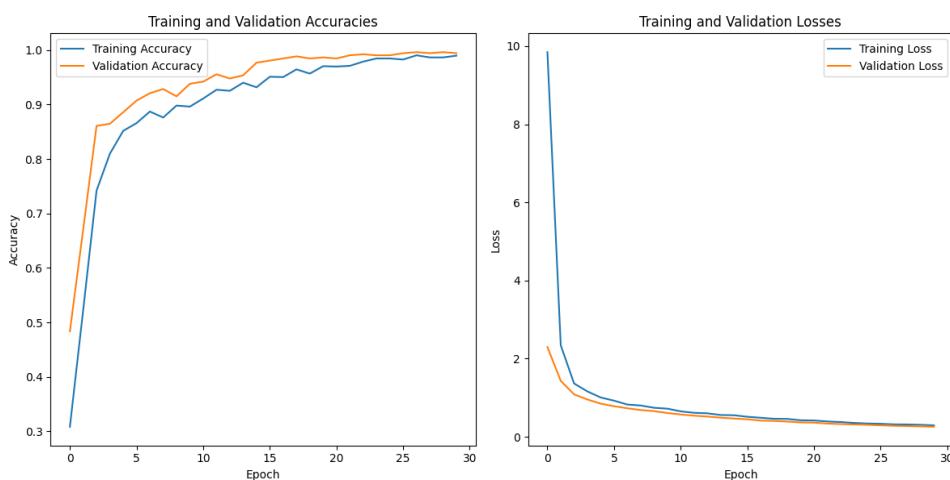
Confusion Matrix for baseline model



Similarly, Figure 78 below shows deep learning models performance over epochs, showing the sharp rise in the training accuracy and the decline in the loss. And a similar rising trend is shown in the validation accuracy which indicates the strong generalization. But a little peak in the validation accuracy, however points to overfitting.

Figure 78

Train and Validation Accuracy & Losses Curves of the Base Model



Evaluation Results After L2 Regularization and Dropout Introduction. When performing regularization, the TensorFlow model utilizes L2 regularization (imported as l2) with the regularization

factor of a 0.001 to address the problems like overfitting by penalizing the big weights by encouraging the smaller, more generalized weight values. The model followed the same structure, using the pre-trained ResNet50 backbone configured without the top layer and ImageNet weights for the feature extraction. The Regularization is explicitly included in the ‘Dense’ layer configurations within both the branches to ensure robustness and generalization. Global average pooling followed by the dense layer with 1024 units and regularization, and the dropout layer to reduce the overfitting.

For metadata branch setup dense layers with 512 and 256 units, each followed by the dropout, all the regularized followed by the similar configurations for the next subsequent layers. Then the concatenation of both branches, from image branch and meta data branch. Output layer with the SoftMax activation and regularization.

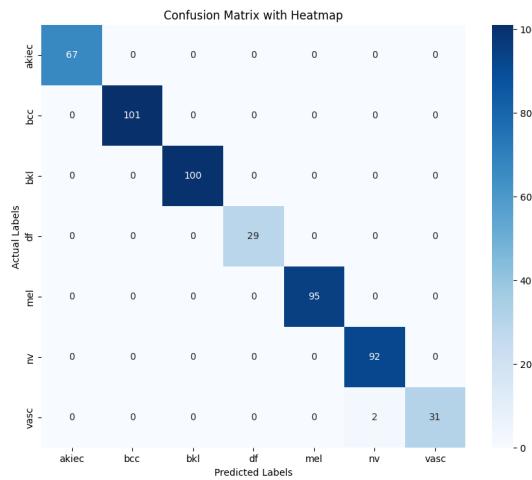
These elements are designed to optimize the model to perform better generalization by penalizing the complexity as seen in the ReNet50 and Metadata branch. This approach is especially aimed for the model's predictive accuracy while preventing it from fitting too closely to the training data.

The improvement in the results is evident after regularization, even though the test loss increased slightly from 0.0683 to 0.1863, the increase in loss did not affect or indicate the failure in overcoming the overfitting, instead it resulted in the more robust and generalized model. The significant improvement in the test accuracy from 97% to 99% and in other metrics like F1-score, precision, and recall also confirms this. The regularization likely helped the model generalize better, reducing the overfitting that might have caused higher performance on training or validation but poorer performance on unseen data.

The confusion matrix shown in Figure 79 informs that the model after the regularization shows the notable improvement in the classification of skin lesions performance of the model which is illustrated in Figure 15. It improvised with strong accuracy in most of the classes. Misclassifications are reduced for 'df', 'mel', and 'nv', indicating slight confusion. Regularization enhanced model generalization capabilities across different skin lesion types, in which by providing clear visualizations of areas of improvement.

Figure 79

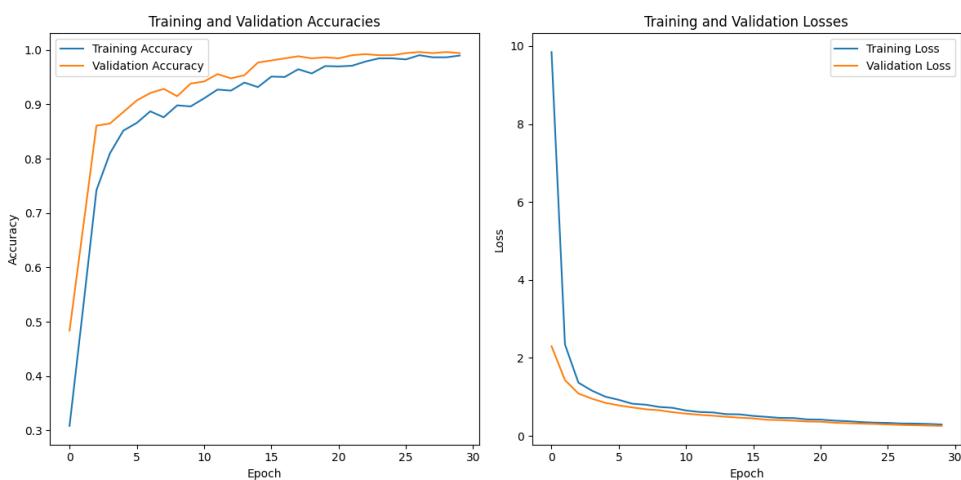
Confusion matrix with heatmap after regularization



The training and validation accuracies as shown in the Figure 80, describes the improvement in the results is evident after regularization, even though the test loss increased slightly from 0.0683 to 0.1863, the increase in loss did not affect or indicate the failure in overcoming the overfitting, instead it resulted in the more robust and generalized model.

Figure 80

ROC and AUC curves after Regularization

**Table 30**

Comparison between the metrics before and after regularization of ResNet50 Model

<i>Metric</i>	<i>Before Regularization</i>	<i>After Regularization</i>
Total Test Loss	0.0683	0.1863
Test Accuracy	97.10%	99.61%
F1 Score	0.9742	0.9962
Precision	97.51%	99.62%
Recall	97.10%	99.61%

Results Comparison of all the Deep Learning Models Implemented

Comparison of the performance of four models (CNN, CNN-OVA, DenseNet121, and ResNet50) implemented for the research is summarized in the Table. A summary of the analysis of the compared key metrics, which were the accuracy, precision, recall, and F1-score, helped gain a more comprehensive understanding of the strengths and weaknesses of each approach.

Table 31

Results Comparison of the deep learning models

Model	Test Accuracy	F1 Score	Precision	Recall
DenseNet121	93.30%	0.9312	0.9123	0.9035
ResNet50	99.61%	0.9962	0.9962	0.9961
CNN - OVA	93.63% (Avg)	0.7892	0.8103	0.6182
CNN	61.70%	0.7157	0.4821	0.617

The achieved performance of the DenseNet121 model by regularization techniques supports the fact that the proper choice of regularization is critical for deep learning models. The base implementation was relatively higher in scores for accuracy and a few other metrics, but the regularized model generalized better since the overfitting problem has been solved. PCA, data augmentation, and small dataset size contributed to the high performance of the model.

PCA reduced dimensionality and noise, highlighting important features so that they could learn better. Consequently, increased data augmentation was done to maintain diversity in the data in a way that will be able to prevent the model from getting very much tuned up on the training set. The results, supported by confusion matrices and precision-recall curves, show that the DenseNet121 model is very powerful for multiclass skin cancer classification. This made the model very balanced with the techniques of regularization, PCA, and data augmentation for effective performance, not only on training data but also outperformed on unseen data, making it a reliable tool for medical diagnostics.

Conclusion

In conclusion, the research project explored Skin Cancer Classification using Deep Learning architectures comprehensively beginning with rigorous data cleaning, missing values imputation for the metadata and further the noisy images were cleaned, where hair was removed using the dull razor method and image transformations, which includes resizing images and applying Principal Component Analysis to reduce dimension and highlight the most relevant features. Data augmentation techniques used were rotation, shifting, and flipping to increase diversity in the dataset and reduce overfitting. The cleaned and transformed dataset is then modeled using CNN, CNN-OVA, ResNet50, and DenseNet121. While all models achieved high accuracy, the ResNet50 model stood out with a near-perfect precision of 0.9962, recall of 0.9961, and F1 scores of 0.9962 following regularization, demonstrating excellent generalization and robustness. Thus, ResNet50 is identified as the best performing model for skin cancer

classification in this study, highlighting the crucial role of regularization techniques in enhancing model performance.

Limitations

The Hardware Constraints include computational resources are limited, and therefore the hardware places an inherent limitation on the size and complexity of the DenseNet121 model. It means that we shall have to use hardware judiciously and make the necessary adjustments to the model settings to ensure its proper working. In addition, the model prediction time must be fast enough for the model to be useful in practical medical scenarios. Extreme caution must be exercised concerning patient privacy and data security throughout the project, and all the rules and regulations must be followed to prevent the leakage of sensitive medical information. Moreover, the performance of the model can be affected by unbalanced data, noisy images, or differences in lighting, which means that some techniques must be used to cope with these issues, ensuring that the model works reliably in various scenarios.

Future Scope

Future works can be oriented in the way of increasing performance and reliability of our skin cancer classification model. Advanced techniques for data augmentation can be implemented. One promising approach toward data augmentation is the use of Generative Adversarial Networks. The use of GANs to produce high-quality synthetic images helps the dataset have more variation compared to traditional ways of augmentation. Such increased variability could help the model to have better generalization from being exposed to a wide range of scenarios. Besides, there is an opportunity to explore further deep learning architectures for enhancement. The currently known to be the best architectures for image classification are EfficientNet, and Vision Transformers. The experiments conducted on these can identify which one maintains the good trade-off between computational efficiency and accuracy in classifying skin cancer. Model performance can further be improved using

ensemble learning. Using bagging, boosting, or stacking, a set of working models with which to supplement one another in order to be stronger and better.

In addition, validation in more clinical situations in real life would help to establish the clinical applicability and provide more reliability of the developed models. Better quality of data, which may be validated, may be accessed in collaboration with dermatologists or any healthcare institute. Furthermore, this model could be integrated into the clinical workflow, and its evaluation performance could be made in live settings so that the practical challenges, area of improvement, and feedback can be taken for classifier improvement. By working in these directions, a more accurate, reliable, and clinically applicable skin cancer detection model would be achieved, which can contribute enormously to early diagnosis and treatment.

References

- Agarwal, K., & Singh, T. (2022). Classification of Skin Cancer Images using Convolutional Neural Networks.
<https://doi.org/10.48550/arXiv.2202.00678>
- Aldeen, Y. M. M. a. N. S. S. a. N. S. S. (2021). Applying transfer Learning using DENSeNeT121 in radiographic image classification: في تصنيف الصور DeNSeNet121 تطبيق التعلم بالنقل باستخدام شبكة: الشعاعية. *Maġallaẗ Al-‘ulūm Al-handasiyyaẗ Wa-al-tiknūlūgiyā Al-ma‘lūmāt*, 5(4), 58–41.
<https://doi.org/10.26389/ajsrp.l060521>
- Ali, A. R. H., Li, J., & Yang, G. (2019). Automating the ABCD rule for melanoma detection. *Computer Methods and Programs in Biomedicine*, 169, 59-69.
<https://doi.org/10.1109/ACCESS.2020.2991034>
- Ayala, A., Figueroa, T. O., Fernandes, B., & Cruz, F. (2021). Diabetic retinopathy improved detection using deep learning. *Applied Sciences*, 11 (24), 11970.
<https://doi.org/10.3390/app112411970>
- Babu, B. P., & Narayanan, S. J. (2022). One-vs-All Convolutional neural networks for synthetic aperture radar target recognition. *Cybernetics and Information Technologies*, 22(3), 179–197.
<https://doi.org/10.2478/cait-2022-0035>
- Brinker, T. J., Hekler, A., Utikal, J. S., Grabe, N., Schadendorf, D., & Klode, J. (2018). Skin Cancer Classification Using Convolutional Neural Networks: Systematic Review. *Journal of Medical Internet Research*, 20, e11936.
<https://doi.org/10.2196/11936>
- Cao, J., et al. (2014). Comprehensive single-cell transcriptional profiling of a multicellular organism. *Science*, 357(6352), 661-667.
<https://doi.org/10.1126/science.1249090>

Chao, X., Liu, Z., Zhao, K., Miao, L., Ma, Y., Zhu, X., Zhou, Q., Wang, S., Li, L., Yang, F., Xu, S., & Chen, H. (2022). An improved transformer network for skin cancer classification. *Computers in Biology and Medicine*, 149, 105939.

<https://doi.org/10.1016/j.combiomed.2022.105939>

Chaturvedi, S. S., Gupta, K., & Prasad, P. S. (2019). Skin Lesion Analyser: An Efficient Seven-Way Multi-Class Skin Cancer Classification Using MobileNet. In A. Hassanien, R. Bhatnagar, & A. Darwish (Eds.), *Advances in Intelligent Systems and Computing*, 981, 123-134.

https://doi.org/10.1007/978-981-15-3383-9_15

Chauhan, T., Palivela, H., & Tiwari, S. (2021). Optimization and fine-tuning of DenseNet model for classification of COVID-19 cases in medical imaging. *International Journal of Information Management Data Insights*, 1 (2), 100020.

<https://doi.org/10.1016/j.jjimei.2021.100020>

Dildar, M., Akram, S., Irfan, M., Khan, H. U., Ramzan, M., Mahmood, A. R., Alsaifi, S. A., Saeed, A. H. M., Alraddadi, M. O., & Mahnashi, M. H. (2021). Skin Cancer Detection: A review using Deep learning techniques. *International Journal of Environmental Research and Public Health*, 18(10), 5479.

<https://doi.org/10.3390/ijerph18105479>

Efeoğlu, E., & Tuna, G. (2022). Machine Learning for Predictive Maintenance: Support Vector Machines and Different Kernel Functions. *Journal of Machinery Manufacture and Reliability*, 51(5), 447–456.

<https://doi.org/10.3103/S1052618822050041>

Elgamal, M. (2013). Automatic skin cancer images classification. *International Journal of Advanced Computer Science and Applications*, 4(3).

<https://doi.org/10.14569/ijacsa.2013.040342>

- Evdokimov, I. V. (2018). Using PERT and Gantt charts for planning software projects on the basis of distributed digital ecosystems. *Journal of Physics Conference Series*, 1074(1), 012127.
<https://doi.org/10.1088/1742-6596/1074/1/012127>
- Fujisawa, Y., Otomo, Y., Ogata, Y., Nakamura, Y., Fujita, R., & Ishitsuka, Y., et al. (2020). Deep-learning-based computer-aided classifiers for diagnosing pigmented skin tumors on small-sized clinical images: a pilot study. *Journal of the American Academy of Dermatology*, 83(1), 91-97.
<https://doi.org/10.1111/bjd.16924>
- Fu'adah, Y. N., Pratiwi, N. C., Pramudito, M. A., & Ibrahim, N. (2020). Convolutional Neural Network (CNN) for Automatic Skin Cancer Classification System. *IOP Conference Series: Materials Science and Engineering*, 982(1), 012005.
<https://doi.org/10.1088/1757-899X/982/1/012005>
- Grandini, M., Bagli, E., & Visani, G. (2020, August 13). Metrics for Multi-Class Classification: an Overview. *arXiv.org*.
<https://arxiv.org/abs/2008.05756>
- Gouda, W., Sama, N. U., Al-Waakid, G., Humayun, M., & Jhanjhi, N. Z. (2022). Detection of skin cancer based on skin lesion images using deep learning. *Healthcare*, 10(7), 1183.
<https://doi.org/10.3390/healthcare10071183>
- Gupta, P., & Mesram, S. (2022). AlexNet and DenseNet-121-based Hybrid CNN Architecture for Skin Cancer Prediction from Dermoscopic Images. *International Journal for Research in Applied Science and Engineering Technology*, 10(7), 540–548.
<https://doi.org/10.22214/ijraset.2022.45325>
- Gururaj, H. L., Manju, N., Nagarjun, A., Manjunath Aradhya, V. N., & Flammini, F. (2023). DeepSkin: A Deep Learning Approach for Skin Cancer Classification. *IEEE Access*, 11.
<https://doi.org/10.1109/ACCESS.2023.3274848>

- Hasan, N., Bao, Y., Shawon, A., & Huang, Y. (2021). DenseNet Convolutional Neural Networks Application for predicting COVID-19 using CT Image. *SN Computer Science/SN Computer Science*, 2(5).
- <https://doi.org/10.1007/s42979-021-00782-7>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 Computer Vision and Pattern Recognition*, 770-778.
- <https://doi.org/10.1109/CVPR.2016.90>
- Hosny, K. M., Kassem, M. A., & Foaud, M. M. (2018). Skin cancer classification using deep learning and transfer learning. *International Biomedical Engineering Conference*
- <https://doi.org/10.1109/cibec.2018.8641762>
- Hosny, K. M., Kassem, M. A., & Foaud, M. M. (2020). Skin Melanoma Classification Using Roi and Data Augmentation with Deep Convolutional Neural Networks. *Multimedia Tools Appl*, 79, 24029–24055.
- <https://doi.org/10.1007/s11042-020-09067-2>
- Huang, G., Liu, Z., Laurens, V. D. M., & Weinberger, K. Q. (2016). Densely connected convolutional networks. *arXiv.org*.
- <https://arxiv.org/abs/1608.06993>
- Kang, H. (2013). The prevention and handling of the missing data. *Korean Journal of Anesthesiology*, 64(5), 402.
- <https://doi.org/10.4097/kjae.2013.64.5.402>
- Kanani, P., & Padole, M. (2019). Deep Learning to Detect Skin Cancer using Google Colab. *International Journal of Engineering and Advanced Technology*, 8(6), 2176-2183.
- <https://doi.org/10.35940/ijeat.F8587.088619>

- Kausar, N., Hameed, A., Sattar, M., Ashraf, R., Imran, A. S., Abidin, M. Z. U., & Ali, A. (2021). Multiclass skin cancer classification using ensemble of Fine-Tuned Deep Learning models. *Applied Sciences*, 11(22), 10593. <https://doi.org/10.3390/app112210593>
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90. <https://doi.org/10.1145/3065386>
- Lee, T., Ng, V., Gallagher, R., Coldman, A., & McLean, D. (1997). Dull Razor: A software approach to hair removal from images. *Computers in Biology and Medicine*, 27(6), 533–543. [https://doi.org/10.1016/s0010-4825\(97\)00020-6](https://doi.org/10.1016/s0010-4825(97)00020-6)
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten Zip code recognition. *Neural Computation*, 1(4), 541–551. <https://doi.org/10.1162/neco.1989.1.4.541>
- Li, H., Pan, Y., Zhao, J., & Zhang, L. (2021). Skin disease diagnosis with deep learning: A review. *Neurocomputing*, 464, 364-393. <https://doi.org/10.1016/j.neucom.2021.08.096>
- Liu, Y., & Yang, S. (2022). Application of Decision Tree-Based Classification Algorithm on Content Marketing. *Journal of Mathematics*. <https://doi.org/10.1155/2022/6469054>
- Madongo, C. T., & Zhongjun, T. (2023). A movie box office revenue prediction model based on deep multimodal features. *Multimedia Tools and Applications*, 82, 31981–32009. <https://doi.org/10.1007/s11042-023-14456-4>

- Manne, R., Kantheti, S., & Kantheti, S. (2020). Classification of Skin cancer using deep learning, Convolutional Neural Networks - Opportunities and vulnerabilities- A systematic Review. *International Journal for Modern Trends in Science and Technology*, 6(11), 101-108.
<https://doi.org/10.46501/IJMTST061118>
- Mishra, S., Imaizumi, H., & Yamasaki, T. (2019). Interpreting Fine-Grained Dermatological Classification by Deep Learning. *CVF Conf ComputeVision Pattern Recognit Workshops*, 0–0.
<https://doi.org/10.1109/CVPRW.2019.00331>
- Pellacani, G., & Seidenari, S. (1998). Comparison between morphological parameters in pigmented skin lesion images acquired by means of epiluminescence surface microscopy and polarized-light videomicroscopy. *Journal of the American Academy of Dermatology*, 39, 215-218.
[https://doi.org/10.1016/s0738-081x\(02\)00231-6](https://doi.org/10.1016/s0738-081x(02)00231-6)
- Perez, F., Avila, S., & Valle, E. (2019). Solo or Ensemble? Choosing a Cnn Architecture for Melanoma Classification. *Conf Comput Vision Pattern Recognit Workshops*.
<https://doi.org/10.1109/CVPRW.2019.00336>
- Polat, K., & Koc, K. O. (2020). Detection of Skin Diseases from Dermoscopy Image Using the Combination of Convolutional Neural Network and One-versus-All. *Journal of Artificial Intelligence and Systems*, 2, 80–97.
<https://doi.org/10.33969/AIS.2020.21006>
- Radwan, N. (2019). Leveraging sparse and dense features for reliable state estimation in urban environments.
<https://doi.org/10.6094/UNIFR/149856>
- Rashid, H., Tanveer, M. A., & Khan, H. A. (2019). Skin Lesion Classification Using GAN-Based Data Augmentation. *International Symposium on Biomedical Imaging*, 916-919.
<https://doi.org/10.1109/ISBI.2019.8759328>

Saeed, J. N., & Zeebaree, S. R. M. (2021). Skin Lesion Classification Based on Deep Convolutional Neural Networks Architectures. *Journal of Applied Science and Technology Trends*, 2(1), 41–51.
<https://doi.org/10.38094/jastt20189>

Sunarya, C., Siswanto, J., Cam, G., & Kurniadi, F. (2023). Skin Cancer Classification using Delaunay Triangulation and Graph Convolutional Network. *International Journal of Advanced Computer Science and Applications*, 14.
<https://doi.org/10.14569/IJACSA.2023.0140685>

Syafa'ah, L., Hanami, I., Rusdia Sofiani, I., & Chasrun, M. (2021). Skin Lesion Image Classification using Convolutional Neural Network. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 6(4), 321-328.
<https://doi.org/10.22219/kinetik.v6i4.1353>

Tschandl, P., Rosendahl, C., & Kittler, H. (2018). The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5(1).
<https://doi.org/10.1038/sdata.2018.161>

Yang, G., Luo, S., & Greer, P. A. (2023). Novel Vision Transformer Model for Skin Cancer Classification. *Neural Process Lett*, 55, 9335–9351.

<https://doi.org/10.1007/s11063-023-11204-5>

Zhao, Z. (2022). Skin Cancer Classification Based on Convolutional Neural Networks and Vision Transformers. *Journal of Physics: Conference Series*, 2405, 012037.
<https://doi.org/10.1088/1742-6596/2405/1/012037>

Appendix B

The code presented in Appendix B shows the progress of the project beginning from data collection through data preparation for modeling.

Figure B1

Connecting to 'ham10000dataset' GCP Bucket

```
|: from google.colab import auth
auth.authenticate_user()
from google.cloud import storage
storage_client = storage.Client()
bucket = storage_client.get_bucket('ham10000dataset')
blob_metadata = bucket.blob('HAM10000_metadata.csv')
local_metadata_location = '/tmp/HAM10000_metadata.csv'
blob_metadata.download_to_filename('local_metadata_location')
```

Figure B2

Reading the Dataset, Imputing Unknown Sex with Model and Imputing Age with RandomForest Regressor

```
import pandas as pd
import numpy as np
df_metadata = pd.read_csv('local_metadata_location')
df_metadata.head()
df_metadata.rename(columns={'dx': 'skin_cancer_type'}, inplace=True)
df_metadata.isnull().sum()
df_metadata.sex.unique()
df_metadata['sex'].replace('unknown', df_metadata['sex'].mode()[0], inplace=True)
df_metadata.isnull().sum()
import matplotlib.pyplot as plt
plt.hist(df_metadata['age'])
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(10, 6))
sns.boxplot(x='skin_cancer_type', y='age', data=df_metadata)
plt.title('Age Distribution by Diagnosis Type')
plt.show()
df_metadata.groupby('skin_cancer_type')['age'].describe()
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestRegressor
label_encoder = LabelEncoder()
for col in ['sex', 'skin_cancer_type']:
    df_metadata[col] = label_encoder.fit_transform(df_metadata[col])
df_notna_train = df_metadata[(df_metadata['age'] != 0) & df_metadata['age'].notna()]
df_na_test = df_metadata[(df_metadata['age']==0) | df_metadata['age'].isna()]
df_na_test.shape
df_notna_train.shape
model = RandomForestRegressor()
model.fit(df_notna_train[['sex', 'skin_cancer_type']], df_notna_train['age'])
predicted_ages = model.predict(df_na_test[['sex', 'skin_cancer_type']])
df_na_test['age'] = predicted_ages
df_na_test.head()
df_metadata_cleaned = pd.concat([df_notna_train, df_na_test])
df_metadata_cleaned.shape
plt.hist(df_metadata_cleaned['age'])
plt.title('Distribution of Age after Imputation - RandomForest')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
df_metadata_cleaned.duplicated().sum()
df_metadata_cleaned.describe()
```

Figure B3

Imputing missing AGE with KNN Imputation

```

import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.impute import KNNImputer

skin_df = pd.read_csv('local_metadata_location')

# Find rows where the 'age' column is 0
zero_age_rows = skin_df.loc[skin_df['age'] == 0]

# Replace '0' values in 'age' column with NaN
skin_df.loc[skin_df['age'] == 0, 'age'] = pd.NA

df_numeric = skin_df[['age']]
df_categorical = skin_df.drop(columns=['age'])

encoder = LabelEncoder()
df_categorical_encoded = df_categorical.apply(encoder.fit_transform)

df = pd.concat([df_numeric, df_categorical_encoded], axis=1)

imputer = KNNImputer(n_neighbors=10)
filled_data = imputer.fit_transform(df)
filled_df = pd.DataFrame(filled_data, columns=df.columns, index=df.index)
# This will only replace the missing values in skin_df with values from filled_df
skin_df = skin_df.where(pd.notna(skin_df), filled_df)

import matplotlib.pyplot as plt
plt.hist(skin_df['age'])
plt.title('Distribution of Age after Imputation - KNN Iputer')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()

```

Figure B4*Removing Hair from Images Using Dull Razor Method*

```

from PIL import Image
import io
Image.open(io.BytesIO(image_bytes))
from google.colab.patches import cv2_imshow
import cv2
# Example DataFrame Loading - ensure it's already loaded
# df_metadata_cleaned = pd.read_csv('path_to_your_dataframe.csv')
# Example process the first image from the DataFrame
blob_path = 'HAM10000_images/'
blob = bucket.blob(blob_path)
# Download the image as bytes
img_bytes = blob.download_as_bytes()
# Convert bytes to a NumPy array
npar = np.frombuffer(img_bytes, np.uint8)
# Decode image
img = cv2.imdecode(npar, cv2.IMREAD_COLOR)
# Dull Razor Method
#GrayScale
grayScale = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY )
#Black hat filter
kernel = cv2.getStructuringElement(1,(9,9))
blackhat = cv2.morphologyEx(grayScale, cv2.MORPH_BLACKHAT, kernel)
#Gaussian filter
bhg= cv2.GaussianBlur(blackhat,(3,3),cv2.BORDER_DEFAULT)
#Binary thresholding (MASK)
ret,mask = cv2.threshold(bhg,10,255,cv2.THRESH_BINARY)
#Replace pixels of the mask
dst = cv2.inpaint(img,mask,6,cv2.INPAINT_TELEA)
#Display images
print('Original Image')
cv2_imshow(img)
print("Gray Scale image")
cv2_imshow(grayScale)
print('Blackhat Filter')
cv2_imshow(blackhat)
print("Binary mask")
cv2_imshow(mask)
print("Clean Processed image")
cv2_imshow(dst)

```

Figure B5*Uploading Cleaned Images to GCP Bucket into a New Folder*

```

for i in range(0,10015):
    blob_path = 'HAM10000_images_clean/' + df_metadata_cleaned.loc[i, 'image_id'] + '.jpg'
    df_metadata_cleaned.loc[i, 'clean_image_path'] = blob_path
import os
os.getcwd()
clean_metadata_blob_path = 'HAM10000_metadata_cleaned.csv'
clean_metadata_blob = bucket.blob(clean_metadata_blob_path)
clean_metadata_blob.upload_from_filename('df_metadata_cleaned.csv')
blob = bucket.blob('HAM10000_metadata_cleaned.csv')
blob.download_to_filename('HAM10000_metadata_cleaned.csv')

```

Figure B6

MinMaxScaler for Standardization of Features and Defining Number of Samples for Data Augmentation

```

import torch
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
skin_df[['age']] = scaler.fit_transform(skin_df[['age']])
# Assuming 'localization' is the column containing categorical variables
skin_df = pd.get_dummies(skin_df, columns=['localization'])
columns = ['localization_abdomen',
           'localization_acral', 'localization_back', 'localization_chest',
           'localization_ear', 'localization_face', 'localization_foot',
           'localization_genital', 'localization_hand',
           'localization_lower extremity', 'localization_neck',
           'localization_scalp', 'localization_trunk', 'localization_unknown',
           'localization_upper extremity']
for col in columns:
    skin_df[col] = skin_df[col].astype(int)
import os

# Use a raw string for the directory path to avoid escape character issues
image_directory = r"cleaned_images/"
image_paths = []

for image_id in skin_df['image_id']:
    # Ensure that the image path uses the correct separator
    image_path = os.path.join(image_directory, image_id + ".jpg")

    image_paths.append(image_path)

skin_df['local_clean_image'] = image_paths
# Define the maximum number of samples per category
max_samples_per_category = 500

# Define a sampling function to sample the desired number of rows from each group
def sample_group(group):
    if len(group) <= max_samples_per_category:
        return group
    else:
        return group.sample(n=max_samples_per_category)

# Group the DataFrame by 'skin_cancer_type' and apply the sampling function to each group
skin_sampled_df = skin_df.groupby('skin_cancer_type', group_keys=False).apply(sample_group)

# Reset the index of the sampled DataFrame
skin_sampled_df.reset_index(drop=True, inplace=True)

# Display the sampled DataFrame
skin_sampled_df.skin_cancer_type.value_counts()

```

Figure B7

Implemented ImageDataGenerator for Data Augmentation

```
from PIL import Image
import numpy as np

def load_images(image_paths):
    images = []
    for path in image_paths:
        # Open the image file
        with Image.open(path) as img:
            images.append(np.array(img))
    # Stack the list of arrays into a single numpy array
    return np.array(images)
images = load_images(skin_sampled_df['local_clean_image'])

from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import numpy as np

# Data augmentation setup
data_generator = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.15,
    zoom_range=0.15,
    horizontal_flip=True,
    fill_mode='nearest'
)
# Fit the data generator
data_generator.fit(images)|
```

Figure B8

PCA For Dimensionality Reduction

```
### PCA

def apply_pca_to_images(images, n_components):

    # Initialize an empty List to store flattened images
    images_flattened = []

    # Flatten each image
    for img in images:
        flattened_image = img.flatten()
        images_flattened.append(flattened_image)

    # Initialize the scaler
    scaler = StandardScaler()

    # Scale the flattened images
    scaled_images = scaler.fit_transform(images_flattened)

    # Stack the scaled images
    images_stacked = np.vstack(scaled_images)

    # Apply PCA
    pca = PCA(n_components=n_components)
    images_pca = pca.fit_transform(images_stacked)

    # Print explained variance ratio
    explained_variance_ratio = pca.explained_variance_ratio_.sum()
    print(f"Total explained variance ratio: {explained_variance_ratio:.2f}")

    return images_pca|
```

```
images_pca = apply_pca_to_images(images, n_components=150)
images_pca
```

Figure B9*Splitting the Data for Training, Validation and Testing*

```

import numpy as np
from sklearn.model_selection import train_test_split

# Labels (assuming `y` is your target array)
y = skin_sampled_df['skin_cancer_type'].values

# First, split the data into a combined training plus validation set and a separate test set
X_temp_img, X_test_img, X_temp_meta, X_test_meta, y_temp, y_test = train_test_split(
    images_pca, skin_sampled_df[cOLUMNS_TO_TRAIN].values, y, test_size=0.2, random_state=42)

# Now, split the temporary training plus validation set into individual training and validation sets
X_train_img, X_val_img, X_train_meta, X_val_meta, y_train, y_val = train_test_split(
    X_temp_img, X_temp_meta, y_temp, test_size=0.25, random_state=42) # Using 0.25 here gives 20% of the original data as validation set, since 0.25 * 0.8 = 0.2

```

Figure B10*Summary of All the Import Packages of Models*

```

from tensorflow.keras.layers import Input, Dense, concatenate, Dropout
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Dense, concatenate, Dropout
from tensorflow.keras.models import Model
from tensorflow.keras import regularizers
import numpy as np
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Dense, Dropout, Flatten, Conv2D, MaxPooling2D, concatenate
from tensorflow.keras.optimizers import Adam
from sklearn.preprocessing import LabelBinarizer
import tensorflow as tf
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.layers import Input, Dense, GlobalAveragePooling2D, concatenate, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.regularizers import l2
from tensorflow.keras.applications import DenseNet121

```

Appendix C

Images presented in Appendix C represent code for the implementation of deep learning models.

Figure C1

MinMaxScaler for Standardization of Features and Defining Number of Samples for Data Augmentation

```

import torch
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
skin_df[['age']] = scaler.fit_transform(skin_df[['age']])
# Assuming 'localization' is the column containing categorical variables
skin_df = pd.get_dummies(skin_df, columns=['localization'])
columns = ['localization_abdomen',
           'localization_acral', 'localization_back', 'localization_chest',
           'localization_ear', 'localization_face', 'localization_foot',
           'localization_genital', 'localization_hand',
           'localization_lower extremity', 'localization_neck',
           'localization_scalp', 'localization_trunk', 'localization_unknown',
           'localization_upper extremity']
for col in columns:
    skin_df[col] = skin_df[col].astype(int)
import os

# Use a raw string for the directory path to avoid escape character issues
image_directory = r"cleaned_images/"
image_paths = []

for image_id in skin_df['image_id']:
    # Ensure that the image path uses the correct separator
    image_path = os.path.join(image_directory, image_id + ".jpg")

    image_paths.append(image_path)

skin_df['local_clean_image'] = image_paths
# Define the maximum number of samples per category
max_samples_per_category = 500

# Define a sampling function to sample the desired number of rows from each group
def sample_group(group):
    if len(group) <= max_samples_per_category:
        return group
    else:
        return group.sample(n=max_samples_per_category)

# Group the DataFrame by 'skin_cancer_type' and apply the sampling function to each group
skin_sampled_df = skin_df.groupby('skin_cancer_type', group_keys=False).apply(sample_group)

# Reset the index of the sampled DataFrame
skin_sampled_df.reset_index(drop=True, inplace=True)

# Display the sampled DataFrame
skin_sampled_df.skin_cancer_type.value_counts()

```

Figure C2

Implemented ImageDataGenerator for Data Augmentation

```
from PIL import Image
import numpy as np

def load_images(image_paths):
    images = []
    for path in image_paths:
        # Open the image file
        with Image.open(path) as img:
            images.append(np.array(img))
    # Stack the list of arrays into a single numpy array
    return np.array(images)
images = load_images(skin_sampled_df['local_clean_image'])

from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import numpy as np

# Data augmentation setup
data_generator = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.15,
    zoom_range=0.15,
    horizontal_flip=True,
    fill_mode='nearest'
)
# Fit the data generator
data_generator.fit(images)|
```

Figure C3

PCA For Dimensionality Reduction

```
### PCA

def apply_pca_to_images(images, n_components):
    # Initialize an empty list to store flattened images
    images_flattened = []

    # Flatten each image
    for img in images:
        flattened_image = img.flatten()
        images_flattened.append(flattened_image)

    # Initialize the scaler
    scaler = StandardScaler()

    # Scale the flattened images
    scaled_images = scaler.fit_transform(images_flattened)

    # Stack the scaled images
    images_stacked = np.vstack(scaled_images)

    # Apply PCA
    pca = PCA(n_components=n_components)
    images_pca = pca.fit_transform(images_stacked)

    # Print explained variance ratio
    explained_variance_ratio = pca.explained_variance_ratio_.sum()
    print(f"Total explained variance ratio: {explained_variance_ratio:.2f}")

    return images_pca|
```

`images_pca = apply_pca_to_images(images, n_components=150)`
`images_pca`

Figure C4*Splitting the Data for Training, Validation and Testing*

```
❶ import numpy as np
from sklearn.model_selection import train_test_split

# Labels (assuming 'y' is your target array)
y = skin_sampled_df['skin_cancer_type'].values

# First, split the data into a combined training plus validation set and a separate test set
X_temp_img, X_test_img, X_temp_meta, X_test_meta, y_temp, y_test = train_test_split(
    images_pca, skin_sampled_df[columnns_to_train].values, y, test_size=0.2, random_state=42)

# Now, split the temporary training plus validation set into individual training and validation sets
X_train_img, X_val_img, X_train_meta, X_val_meta, y_train, y_val = train_test_split(
    X_temp_img, X_temp_meta, y_temp, test_size=0.25, random_state=42) # Using 0.25 here gives 20% of the original data as validation set, since 0.25 * 0.8 = 0.2
```