COMPARATIVE ANALYSIS OF TRANSFER LEARNING MODELS FOR AYURVEDA DOSHA CLASSIFICATION

A PROJECT REPORT

Submitted by

KAUSHIK HARIHARAN G

(2023176027)

A report for the phase-I of the project
submitted to the Faculty of

INFORMATION AND COMMUNICATION ENGINEERING

in partial fulfillment
for the award of the degree
of

MASTER OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

SPECIALIZATION IN AI & DS

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING, GUINDY
ANNA UNIVERSITY
CHENNAI 600 025

JAN 2025

ANNA UNIVERSITY

CHENNAI - 600 025

BONAFIDE CERTIFICATE

Certified that this project report titled **COMPARATIVE ANALYSIS OF TRANSFER LEARNING MODELS FOR AYURVEDA DOSHA CLASSIFICATION** is the bonafide work of **KAUSHIK HARIHARAN G (2023176027)** who carried out project work under my supervision. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

PLACE:CHENNAI
DATE:

Dr. S. SENDHIL KUMAR
PROFESSOR
PROJECT GUIDE
DEPARTMENT OF IST, CEG
ANNA UNIVERSITY
CHENNAI 600025

COUNTERSIGNED

Dr. S. SWAMYNATHAN
HEAD OF THE DEPARTMENT
DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING, GUINDY
ANNA UNIVERSITY
CHENNAI 600025

ABSTRACT

Ayurveda, an ancient system of medicine from India, focuses on balancing the body's energies—Vata, Pitta, and Kapha—to maintain health and wellness. The integration of Artificial Intelligence (AI) and Machine Learning (ML) in traditional healthcare is gaining traction, as it can provide personalized health assessments and assist in complex diagnostic processes. However, the subjective nature of Dosha identification and the limited availability of structured data present significant challenges. To address these, this study proposes the use of deep learning techniques, particularly transfer learning, to classify Dosha types from images of hair, skin, face shape, eyes, and lips, each labeled according to Ayurvedic principles.

By fine-tuning models like VGG16, VGG19, ResNet50, DenseNet121, Inception V3, EfficientNet, and MobileNet, this research aims to bridge traditional Ayurvedic practices with modern AI technologies, offering a more accurate and accessible means of Dosha identification. The methodology involved applying transfer learning to adapt pre-trained models to the specific task of Dosha classification. A comparative analysis was conducted by fine-tuning these models on labeled datasets to evaluate their ability to recognize patterns associated with the three Dosha types. Performance was evaluated using standard metrics such as accuracy, precision, and recall.

Models such as MobileNet and DenseNet121 demonstrated superior performance, highlighting their efficiency in classifying Dosha types across diverse datasets. This study emphasizes the integration of AI with Ayurveda, paving the way for developing real-time diagnostic systems to support personalized healthcare solutions.

ABSTRACT (TAMIL)

ACKNOWLEDGEMENT

It is my privilege to express my deepest sense of gratitude and sincere thanks to **Dr. S. SENDHIL KUMAR ,** Professor , Department of Information Science and Technology, College of Engineering, Guindy, Anna University, for his constant supervision, encouragement, and support in my project work. I greatly appreciate the constructive advice and motivation that was given to help me advance my project in the right direction.

I am grateful to **Dr. S. SWAMYNATHAN,** Professor and Head, Department of Information Science and Technology, College of Engineering Guindy, Anna University for providing us with the opportunity and necessary resources to do this project.

I would also wish to express my deepest sense of gratitude to the Members of the Project Review Committee: **Dr. S. SRIDHAR,** Professor, **Dr. G. GEETHA,** Associate Professor, **Dr. D. NARASHIMAN,** Teaching Fellow Department of Information Science and Technology,College of Engineering Guindy, Anna University, for their guidance and useful suggestions that were beneficial in helping me improve my project.

I also thank the faculty member and non teaching staff members of the Department of Information Science and Technology, Anna University, Chennai for their valuable support throughout the course of our project work.

KAUSHIK HARIHARAN G

2023176027

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

AI: Artificial Intelligence

CNN: Convolutional Neural Network

DenseNet: Densely Connected Convolutional Networks

DL: Deep Learning

FN: False Negative

FP: False Positive

HCI: Human-Computer Interaction

IoU: Intersection over Union

ML: Machine Learning

PIL: Python Imaging Library

RDC: Rounded Double Chin

RGB: Red, Green, Blue

ReLU: Rectified Linear Unit

ResNet: Residual Network

ROC: Receiver Operating Characteristic

SGD: Stochastic Gradient Descent

TA: Thin Angular

TL: Transfer Learning

TN: True Negative

TP: True Positive

TT: Tapering Triangular

VGG: Visual Geometry Group

VPK: Vata, Pitta, Kapha

CHAPTER 1

INTRODUCTION

1.1 AYURVEDA AS TRADITIONAL MEDICINE AND ITS IMPORTANCE

Ayurveda, one of the oldest holistic medicine systems, originates from ancient India and focuses on achieving harmony between the body, mind, and spirit. It emphasizes the balance of three fundamental energies, known as doshas—Vata, Pitta, and Kapha—that govern physiological and mental processes. Each person is believed to have a unique combination of these doshas, and maintaining their balance is crucial for health and well-being. Ayurveda is not only used for treating illnesses but also plays a preventive role in maintaining optimal health. It has gained widespread recognition across the globe, particularly in lifestyle management and preventive healthcare, offering personalized solutions through diet, meditation, herbal treatments, and physical therapies. The significance of Ayurveda lies in its holistic approach, which considers the individual as a whole and seeks to prevent diseases before they arise.

In addition to its preventive focus, Ayurveda also offers personalized therapeutic interventions based on an individual's unique constitution, or Prakriti, which is determined by the dominant dosha or combination of doshas. These personalized approaches can help address a wide range of health conditions, from chronic diseases to emotional imbalances, by restoring harmony within the body. The growing interest in Ayurveda worldwide highlights its potential to complement modern medical practices, providing a more holistic and individualized approach to health management.

1.2 APPLICATION OF DEEP LEARNING IN THE FIELDS OF HEALTHCARE

Deep learning, a subset of artificial intelligence (AI), has had a profound impact on healthcare, particularly in medical image analysis, disease diagnosis, and personalized treatment planning. The use of Convolutional Neural Networks (CNNs), a deep learning technique, has revolutionized fields such as radiology, pathology, and dermatology, enabling machines to identify patterns and anomalies in medical images with high accuracy. Deep learning models are capable of processing large volumes of data, uncovering hidden patterns that may not be easily visible to the human eye, and assisting healthcare professionals in making faster and more accurate diagnoses. In addition to image analysis, deep learning has also been applied in predictive modeling, drug discovery, and the development of personalized health management systems. As the healthcare industry continues to adopt AI technologies, the potential for deep learning to enhance diagnostic precision, treatment efficacy, and patient outcomes is vast.

Moreover, deep learning is also being utilized to streamline healthcare workflows and improve operational efficiency. For example, AI-powered systems can automate administrative tasks like patient data entry, appointment scheduling, and medical billing, reducing the burden on healthcare professionals and allowing them to focus more on patient care. Additionally, deep learning models can support decision-making by analyzing a patient's medical history, lab results, and genetic information, leading to more accurate and timely interventions. The ability to integrate deep learning with various healthcare data sources holds great promise for the future of personalized medicine and predictive healthcare.

1.3 CHALLENGES

1.3.1 Data Collection and Annotation

The collection and annotation of Ayurvedic data presents a significant challenge due to the domain-specific nature of Ayurveda. There is a limited availability of publicly accessible datasets online, making it necessary to scrape data and manually annotate it. The annotation process requires a deep understanding of Ayurvedic principles and conditions, as each data point must be accurately labeled according to the unique dosha attributes that correspond to specific Ayurvedic conditions. This process demands careful attention to detail and domain expertise to ensure the correct classification of dosha imbalances, which can be complex and subjective. Without proper annotations, the quality of the dataset and the performance of the model could be compromised.

1.3.2 Imbalanced Datasets

Even when datasets are organized and labeled, achieving balance remains a significant challenge. Certain dosha types or conditions may be underrepresented in the dataset, making it difficult to prepare a well-balanced dataset for model training. This imbalance can lead to biased predictions, as the model may overfit to the more frequent classes while underperforming on the less frequent ones. During validation, this imbalance causes the model to yield inaccurate results, which can diminish the overall performance of the model. Proper handling of imbalanced datasets, such as through data augmentation or re-sampling techniques, is essential for achieving more reliable and robust model predictions.

1.3.3 Resource Constraints and Scalability

Resource constraints, both in terms of computational power and available datasets, often hinder the scalability of deep learning models in healthcare applications. Training deep learning models, especially on large and complex datasets, requires significant computational resources, including high-performance GPUs and large memory capacities. Additionally, the lack of extensive, well-annotated Ayurvedic datasets limits the ability to scale these models effectively. As the system grows and more data is incorporated, ensuring that the infrastructure can handle the increased load without compromising performance becomes crucial. Without adequate resources, the scalability of the system can be severely limited, preventing it from being deployed in real-world, large-scale applications.

1.4 PROBLEM STATEMENT

Current healthcare systems primarily rely on biomedical parameters such as heart rate, blood pressure, and physical activity to assess health, often overlooking traditional Ayurvedic practices that emphasize the balance of the three doshas—Vata, Pitta, and Kapha—affecting both physical and mental well-being. While Ayurvedic dosha analysis offers a holistic approach to health, there is a lack of technological integration to classify doshas based on physical attributes such as hair type, skin condition, lip characteristics, eye shape, and face shape, which are closely tied to dosha imbalances. The challenge lies in developing deep learning models capable of accurately classifying dosha types using image-based data from these attributes, as existing systems do not provide a reliable way to assess dosha imbalances through visual cues, hindering the ability to combine traditional Ayurvedic knowledge with modern healthcare practices.

1.5 PROPOSED SOLUTION

The proposed solution integrates deep learning models with Ayurvedic Dosha classification to provide personalized health assessments. Leveraging transfer learning techniques, the system utilizes pre-trained models such as VGG16, VGG19, ResNet-50, MobileNet, InceptionV3, and DenseNet121 to classify Dosha imbalances based on image data from multiple body features, including skin, hair, eyes, lips, and face shape. A comparative analysis of these models will be conducted across five diverse datasets to identify the best-performing architecture, ensuring high accuracy and robustness in Dosha identification. Data augmentation techniques are employed to enhance model generalization, enabling better performance on unseen data while addressing dataset limitations by creating a diverse and enriched sample set. This holistic approach bridges modern AI with traditional Ayurveda, offering a scalable and efficient system for real-time Dosha tracking, empowering individuals to monitor their imbalances and make informed health decisions aligned with personalized Ayurvedic recommendations.

1.6 OBJECTIVE

The primary objective of this project is to develop a deep learning-based system for Ayurvedic dosha classification using image data. The system aims to bridge the gap between traditional Ayurvedic knowledge and modern healthcare practices by leveraging advanced deep learning techniques, particularly transfer learning. The specific objectives of this research are:

1) Data Collection/Creation, Including Augmentation: To gather diverse datasets that cover various body features, such as hair, skin, eyes, lips, and face shape, all relevant to Dosha classification. This will include creating or collecting sufficient samples, followed by data augmentation to enhance the

diversity and quality of the dataset, ensuring better generalization for model training.

2) Design/Develop AI/ML Models for Image-Based Dosha Classification: To develop deep learning models utilizing transfer learning techniques to classify Dosha imbalances from image data. Pre-trained models such as VGG16, VGG19, ResNet-50, MobileNet, InceptionV3, and DenseNet121 will be adapted for the task, optimizing them to perform robustly in Ayurvedic Dosha classification.

3) Comparative Study of the Seven Models: To conduct a comprehensive evaluation of the performance of the seven deep learning models in classifying Dosha imbalances. The comparative study will assess accuracy, precision, recall, and F1 score across various datasets to identify the most efficient and reliable model for Dosha classification.

1.7 SCOPE OF THE PROJECT

The project focuses on integrating deep learning techniques with Ayurvedic dosha classification using image-based data. It involves five datasets that capture physical attributes, such as hair type, skin condition, lip characteristics, eye shape, and face shape, which are associated with dosha imbalances. Pre-trained models, including VGG16, VGG19, ResNet-50, MobileNet, InceptionV3, and DenseNet121, will be fine-tuned using transfer learning to classify dosha types. Data augmentation will enhance model generalization, and performance will be evaluated with accuracy, precision, recall, and F1-score. The system aims to provide personalized Ayurvedic health assessments based on visual attributes, serving as a complementary tool for Ayurvedic health management.

1.8 ORGANIZATION OF THE REPORT

This section provides an overview of the project structure, outlining the sequence and content of each chapter. It serves as a roadmap for navigating through the project and understanding its progression.

Chapter 2: This chapter provides a literature summary focusing on key topics such as data collection, data augmentation, Ayurveda, and transfer learning. It highlights the current state of research and practices in these areas, with a particular emphasis on their applications in the project.

Chapter 3: This chapter covers the system architecture, detailing the processes followed by each module, including datasets and the workflow of the project. It explains the structure of the system and the integration of various components for efficient classification.

Chapter 4: This chapter outlines the implementation of the project, which includes web scraping, data preprocessing, data augmentation, model training, and evaluation. It provides a detailed explanation of the steps taken to prepare the data, train the models, and assess their performance.

Chapter 5: This chapter focuses on result analysis, which involves selecting the best-performing models. It presents a comparison of the models using graphical representations such as bar charts and comparison graphs, showcasing key metrics such as accuracy, loss, precision, recall, and F1 score.

Chapter 6: This chapter summarizes the findings and discusses the significance of the work. It draws conclusions based on the results and highlights the implications of the project. It also outlines possible future enhancements and areas for further research.

CHAPTER 2

LITERATURE SURVEY

The literature survey focuses on the application of machine learning and deep learning techniques in diverse fields, including healthcare, Ayurveda, medicinal plant identification, and lung cancer detection. It examines studies leveraging convolutional neural networks (CNNs) and transfer learning models such as ResNet50, VGG16, MobileNet, and InceptionV3 for tasks like COVID-19 and pneumonia detection, image classification, and Ayurvedic Dosha prediction. Key highlights include advancements in AI for early diagnosis, improvements in classification accuracy, and the significance of hyperparameter tuning in enhancing model performance. These findings underscore the transformative role of AI in healthcare and its potential to address challenges across various domains..

2.1 EXISTING SYSTEMS

Existing systems predominantly rely on machine learning techniques and text-based datasets, with only a few utilizing image datasets. Even when images are used, deep learning methods in these systems often lack efficiency and generalization. These approaches typically focus on analyzing a single feature or body part, which limits their effectiveness in accurately identifying Doshas.

Dosha identification requires a holistic analysis of various body features, as individual parts can sometimes vary in their representation of Doshas. For instance, hair type may lead to misidentification, while features like eyes, lips, and face shape remain consistent and provide more reliable indicators.

Comprehensive analysis, incorporating images of hair, skin, lips, eyes, face shape, and additional features, is essential for accurate Dosha identification. Existing systems, being dependent primarily on machine learning models, fail to address this multidimensional requirement effectively.

2.1.1 Data Collection

Steven Euijong Whang et al .[1]reviewed data-centric AI techniques to improve data quality for robust and fair model development. They explored methods across data collection, validation, cleaning, integration, and robust training, emphasizing their convergence. Key findings included the integration of data cleaning with robust training, combining data validation with fairness measures, and incorporating fairness and robustness into data collection and model training. They highlighted hybrid approaches like SELFIE and DivideMix for noisy labels and stressed the importance of trustworthy AI by addressing fairness, robustness, privacy, and explainability. The study concluded that as data-centric AI evolves, the convergence of these techniques will be essential for building effective and ethical AI systems.

Hamed Taherdoost et al .[2], provides a comprehensive overview of data collection methods and ethical considerations for research studies. The article highlights the significance of data collection in answering research questions and categorizes various methods, including questionnaires, interviews, focus groups, observations, surveys, case studies, and experimental approaches. It discusses the advantages and disadvantages of these methods and addresses the challenges researchers face in selecting the most suitable technique based on their data type. Ethical considerations are emphasized, such as ensuring participant confidentiality, obtaining informed consent, and minimizing risks. Additional aspects, like the complexities of ethical clearance processes, international regulations, and data protection, are also explored. The

article concludes by underscoring the importance of aligning ethical practices with effective data collection to enhance the quality and reliability of research outcomes.

Yuji Roh et al .[3], explores the challenges of data collection in machine learning. It highlights issues such as the lack of labeled data and the need for large datasets in deep learning. The article discusses various data acquisition and labeling methods, including manual, crowdsourcing, and automated techniques, as well as approaches like active learning and weak supervision. Transfer learning is presented as a solution for training on smaller datasets. Key challenges include selecting sufficient and appropriate data, balancing accuracy and scalability, and addressing crowdsourcing and labeling quality control. Future research directions include improving metadata extraction, model fairness, and comparing labeling techniques based on application-specific factors.

Dipali Shete et al .[4] explore web data extraction and classification for e-commerce websites, highlighting the challenges of dealing with diverse and unstructured content. They compare classification techniques like Support Vector Machine (SVM), Decision Trees, Naive Bayes, and K-Nearest Neighbors (KNN), showing that methods using multilayer perceptrons (K2) and Random Forest (K4) offer the highest accuracy, with K4 achieving 98%. The paper concludes that neural networks and decision trees perform better even with noisy data, but their effectiveness diminishes as the amount of content increases, revealing scalability issues.

Ajay Sudhir Bale et al .[5]explore various web scraping techniques and their effectiveness on modern websites. The study evaluates seven scraping methods, including basic techniques using the Python requests library and advanced methods with Selenium and the undetectable-chromedriver library,

testing them across 15 top websites in categories like e-commerce, education, real estate, and more. The authors highlight that bots, responsible for half of web traffic, can be damaging, especially through database scraping. They measured key parameters such as the number of requests, the time before detection, and the success of data extraction to assess the performance of each approach. The results reveal that websites with dynamic content, often rendered through JavaScript, are more vulnerable to scraping, while the use of Selenium in headless mode proves effective for handling such content. Additionally, the undetectable-chromedriver library showed promise in bypassing advanced anti-bot protections. The paper concludes that while bot detection is improving, there is still significant room for enhancement in protecting websites from scraping attacks. Further research is encouraged, with suggestions to expand the tests across more websites, add additional parameters, and develop automated systems to better assess data extraction success.

Vidhi Singrodia et al .[6] explore web scraping in their paper. The paper highlights the significance of web scraping in collecting unstructured data from the internet, which is essential for data analysis and processing. It reviews various tools and software for web scraping, explaining their operating principles, strengths, and limitations. The paper also discusses the diverse applications of web scraping in fields such as Big Data, Business Intelligence, Open Government Data, and the development of innovative applications, emphasizing its relevance for both B2B and B2C systems.

2.1.2 Data Augmentation

Connor Shorten et al .[7] discuss the challenge of overfitting in deep convolutional neural networks that require large datasets, particularly in domains like medical image analysis with limited data. The paper focuses on Data Augmentation techniques to expand training datasets and improve model

performance, covering methods like geometric transformations, color space adjustments, kernel filters, random erasing, and generative adversarial networks (GANs). The authors also explore other aspects like test-time augmentation, resolution impact, and curriculum learning. Although Data Augmentation can't address all biases, it helps reduce issues like lighting, occlusion, and scale variations, preventing overfitting in data-limited scenarios.

Kiran Maharana et al . [8] review data pre-processing and augmentation techniques in machine learning. The paper discusses challenges like noise, missing, and inconsistent data, emphasizing the importance of pre-processing steps such as cleaning, integration, and transformation to improve data quality. It highlights data augmentation methods like flipping, rotation, and noise injection to reduce overfitting and enhance model performance, especially with limited data. The authors also explore pre-processing techniques such as handling missing values, fixing structural errors, and feature extraction, stressing the need for automation in operational data preparation.

Teerath Kumar et al .[9] provide a comprehensive survey on image data augmentation techniques to mitigate overfitting in deep learning models. The paper introduces a novel taxonomy of augmentation methods and evaluates their impact on image classification, object detection, and semantic segmentation. It identifies challenges, including the lack of label smoothing in image manipulation, the need for importance-based augmentation, and the impact of mixing multiple salient parts in image augmentation. The authors discuss the potential of GANs and VAEs for generating synthetic data and suggest future research on optimal augmentation sequences, data augmentation in feature space, and systematic approaches for selecting techniques based on dataset and model characteristics. The survey concludes by emphasizing data augmentation's role in improving model performance and generalization.

Puneet Kumar et al .[10] the importance of pre-processing techniques in enhancing CNN performance for image recognition tasks. The paper examines various methods like data augmentation, normalization, resizing, and noise reduction, highlighting their impact on CNN performance. The authors analyze the effect of pre-processing on machine learning algorithms applied to Twitter datasets for emotion recognition, focusing on SVM, Maximum Entropy, and Naïve Bayes algorithms. The study shows that pre-processing significantly improves model accuracy, particularly for larger datasets, and emphasizes its crucial role in data analysis. The conclusion highlights pre-processing's essential role in ensuring data quality and robustness, enhancing machine learning and deep learning models.

Agnieszka Mikołajczyk et al .[11] explore data augmentation techniques to improve deep learning for image classification, addressing challenges like insufficient data and class imbalance. The paper compares methods such as image transformations, Style Transfer, and GANs, with a focus on using style transfer to generate high-quality images by merging content from one image with the style of another. Validated on medical datasets, the paper finds traditional methods quick but limited, while advanced methods like GANs offer more diversity but are computationally expensive. The authors suggest combining various techniques to enhance data-hungry models, with future work testing neural networks pre-trained on synthetic images from style transfer.

2.1.3 Ayurveda

Gayatri Gadre et al .[12] proposes a system that integrates advancements in computer vision and image processing with Ayurveda, an ancient Indian medical science. Rooted in the belief that the universe and humans are composed of five elements, Ayurveda categorizes individuals into three Doshas—Vata, Pitta, and Kapha—based on their unique physiological

and psychological traits. Traditionally, Ayurvedic practitioners rely on manual assessments of physical features and pulse for diagnosis and personalized treatment. This system seeks to automate *Prakruti* analysis by examining facial features such as hair, eyes, and skin color. Despite challenges like variations in human faces due to race, age, and lighting conditions, initial efforts in applying these technologies have shown promising results. To overcome the challenge of creating a labeled dataset for Ayurvedic *Prakruti* categories, a combination of manual image collection and existing facial recognition datasets was utilized. This research represents a pioneering effort to enhance diagnostic tools and personalized healthcare solutions within Ayurveda. Future exploration of deep learning techniques and larger datasets could further improve the accuracy of *Prakruti* classification, potentially transforming traditional Ayurvedic practices.

Vishu Madaan et al .[13]proposed a system that applies machine learning techniques to the field of Ayurveda, an ancient Indian medical science aimed at achieving a harmonious balance of body, mind, and soul for optimal health. Their research highlights the necessity for standardized tools to evaluate individual constitution (*Prakruti*) and imbalances related to the three Doshas—Vata, Pitta, and Kapha—and their psychological impacts on health.The proposed methodology employs various machine learning models, including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Artificial Neural Networks (ANN), utilizing data collected via a specialized questionnaire. To enhance model performance, hyperparameter tuning is applied. Among the tested algorithms, SVM proved effective for high-dimensional data, while ensemble methods such as XGBoost and CatBoost significantly improved classification accuracy. After hyperparameter optimization, the CatBoost model achieved an accuracy of 95%. This approach demonstrates the potential of machine learning to accurately predict Ayurvedic body constituents, offering individuals a valuable resource for understanding Dosha imbalances and enhancing self-awareness and health management

without requiring direct consultation with Ayurvedic practitioners.

Sanyam Jain et al .[14] explore the use of machine learning and deep learning techniques to identify Ayurvedic Doshas (Vata, Pitta, Kapha) based on facial and body features. The paper focuses on *Darshana*, a visual diagnosis method, where body build, face shape, skin type, eyes, and hair are analyzed. Algorithms like principal component analysis, k-means clustering, and convolutional neural networks (CNNs) classify these features to determine the dominant Dosha. Future work includes testing this approach with real datasets and developing a mobile or web application for Dosha analysis to promote healthier lifestyles and early disease prevention.

Lakshmi Bheemavarapu et al .[15] review the role of Data Science in Ayurveda-based disease diagnosis using *Prakruti* type in *Trividha Pariksha*. The paper discusses how machine learning and deep learning techniques are enhancing the speed and accuracy of diagnosis in Ayurveda by automating the identification of *Prakruti* types and diseases. While current AI models automate one or two phases of *Trividha Pariksha*, the paper emphasizes the importance of automating all three phases for more accurate results. Additionally, the review highlights the potential for developing models for disease identification and treatment, calling for further research in these areas to improve Ayurveda diagnostics.

Pradeep Tiwari et al .[16]explore the use of machine learning to predict Ayurvedic constitution types, known as *Prakruti*, based on phenotypic traits. The study analyzed data from 147 healthy individuals of three extreme *Prakruti* types in a genetically homogeneous population in Western India. Using unsupervised and supervised machine learning approaches, the researchers successfully classified individuals according to their *Prakruti* types, identifying distinct clusters corresponding to the three classes.The study demonstrates that

Prakruti types form verifiable clusters within a multidimensional space of interrelated phenotypic traits, providing a framework for predicting *Prakruti* from these attributes. This approach could aid in precision medicine, helping to stratify endophenotypes in both healthy and diseased populations. The study further emphasizes that clinical *Prakruti* evaluation can be formalized through machine learning, offering a more empirical method that could assist trained Ayurveda physicians in making accurate predictions.

Rpreet Singh et al .[17] conducted a study on Ayurveda medicine, focusing on chronic diseases treated through telemedicine. Data from 353,000 patients was captured over teleconsultations by around 300 Ayurveda doctors. They found that 82% of patients had diseases chronic for over a year, with digestive issues being the most prevalent (30.6%). The study showed that Ayurveda treatment provided relief in over 76% of cases, despite the chronic nature of the diseases. The analysis revealed age- and sex-based disease patterns and highlighted the efficacy of Ayurveda for chronic, non-life-threatening diseases. The study also showed that Ayurveda treatment could help reduce healthcare costs if applied early before diseases become chronic.

2.1.4 Transfer Learning

Sarra Guefrechi et al .[18] propose a system that leverages deep learning advancements for medical image analysis in the context of the COVID-19 pandemic. Their research highlights the effectiveness of pre-trained convolutional neural networks (CNNs) such as ResNet50, InceptionV3, and VGG16 in classifying chest X-ray images as either COVID-19 positive or normal, achieving impressive accuracies of 97.20%, 98.10%, and 98.30%, respectively. By employing data augmentation techniques like random rotation and noise addition, the robustness of the models was enhanced. The methodology involves preprocessing images and utilizing transfer learning,

where the upper layers of the models are fine-tuned while preserving the learned weights of the lower layers. This approach has proven beneficial for rapid diagnosis, aiding healthcare professionals in timely clinical decision-making. Future research may explore the combination of multiple models to further improve diagnostic accuracy, emphasizing the transformative potential of deep learning in healthcare during crises.

Sheldon Mascarenhas et al .[19] proposed a system that leverages recent advancements in artificial intelligence (AI) and machine learning to enhance image classification tasks using Convolutional Neural Networks (CNNs) like VGG16, VGG19, and ResNet50. These models are adept at extracting features and recognizing patterns in images, significantly outperforming traditional artificial neural networks (ANNs). Their study evaluates these architectures on a dataset of 6,000 images across five categories—shoes, beauty products, jewelry, watches, and bags—finding that ResNet50 achieved the highest accuracy. This research highlights the potential of CNNs in diverse applications, ranging from consumer sectors to healthcare, where improved accuracy in disease detection can be life-saving. It underscores the transformative role of deep learning in advancing image classification systems across various domains.

Kayiram Kavitha et al .[20] proposed a system for the automated classification of medicinal plant species, leveraging advancements in deep learning and computer vision. Their approach addresses the limitations of traditional identification methods, which often rely on expert knowledge and can be both time-consuming and error-prone. By employing Convolutional Neural Networks (CNNs) such as ResNet50, MobileNet, Xception, DenseNet121, and Inception v3, they enhance the accuracy and efficiency of plant classification. Specifically, ResNet50 utilizes batch normalization and residual learning to mitigate issues related to vanishing gradients, while DenseNet121 promotes

better feature propagation through its interconnected layers. Xception's design focuses on improving efficiency via depthwise separable convolutions, and MobileNet is optimized for lightweight applications. Notably, their system incorporates Inception v3, which employs convolution factorization and auxiliary classifiers, achieving a remarkable classification accuracy of 95.16% for Indian medicinal plants. Their research highlights the transformative potential of artificial intelligence in revolutionizing medicinal plant identification, offering significant benefits to the fields of botany and pharmacognosy.

Rachna Jain et al .[21]proposed a system for the early detection of pneumonia in children through the use of x-ray imaging, which is crucial given its status as a leading cause of child mortality worldwide, particularly in South Asia and Sub-Saharan Africa. Their research employs Convolutional Neural Networks (CNNs) to classify x-ray images into pneumonia and non-pneumonia categories. They evaluate six different models, including custom architectures and pre-trained models such as VGG16, VGG19, ResNet50, and Inception-v3. Notably, their Model 2, featuring three convolutional layers, achieved a validation accuracy of 92.31% and a recall rate of 98%, outperforming other models, including VGG19. While ResNet50 and Inception-v3 exhibited significant overfitting, VGG16 and VGG19 demonstrated superior performance with lower overfitting rates. These results highlight the effectiveness of CNNs, particularly Model 2 and VGG19, in facilitating real-time pneumonia detection, emphasizing the potential for future advancements in enhancing diagnostic capabilities.

Christian Szegedy et al . [22] reimagined the Inception architecture to improve efficiency and scalability in computer vision[22]. Their Inception-v3 model achieved 21.2% top-1 and 5.6% top-5 error on the ILSVRC 2012 classification challenge, using only 25 million parameters and 5 billion

multiply-adds. This approach, incorporating factorized convolutions and regularization techniques, significantly reduced computational costs while maintaining high accuracy. An ensemble of four models further lowered the top-5 error to 3.5%, setting a new standard for efficient, high-performance vision networks.

Gao Huang et al .[23] a novel architecture where every layer connects to all preceding layers, enabling efficient feature reuse and strengthened feature propagation. Unlike traditional convolutional networks with one connection per layer, DenseNets establish $L(L+1)/2$ connections for L layers, effectively addressing the vanishing gradient problem. These networks are highly parameter-efficient, using fewer parameters than ResNets while achieving comparable or better accuracy. DenseNets also enable implicit deep supervision, where each layer benefits from supervision through shorter connections, and their ability to reuse features results in compact models with reduced redundancy. They have achieved state-of-the-art results on benchmarks like CIFAR-10, CIFAR-100, SVHN, and ImageNet, requiring less computation than other architectures. With their compact and accurate design, DenseNets are highly suitable for tasks requiring feature extraction, making them valuable for various computer vision applications.

Jian sun et al .[24] introduces a residual learning framework to address the challenges of training very deep neural networks. By reformulating layers to learn residual functions $F(x) = H(x) - x$, the framework simplifies optimization and mitigates performance degradation in deep networks. The study demonstrates the feasibility of training networks as deep as 152 layers, outperforming traditional architectures like VGG with fewer parameters and reduced computational requirements. Residual networks (ResNets) achieved state-of-the-art performance on ImageNet during ILSVRC 2015, with a top-5 error of 3.57%, excelling across tasks like classification, detection, and

localization. Key innovations include shortcut connections that improve backpropagation efficiency without adding overhead, showing consistent gains on datasets like ImageNet and CIFAR-10. ResNet-152, for instance, achieved better accuracy and efficiency with an $8\times$ deeper architecture than VGG-19. These advancements have made residual networks a cornerstone in computer vision, inspiring further innovations in neural network design across diverse applications.

Andrew G. Howard et al.[25] introduces MobileNets, a lightweight model architecture designed for mobile and embedded vision applications. Using depthwise separable convolutions, MobileNets offer a streamlined design with two global hyperparameters that balance latency and accuracy, allowing model builders to select the right size based on application constraints. Extensive experiments show MobileNets outperform other models on ImageNet classification, demonstrating strong performance in object detection, fine-grain classification, face attribute recognition, and geo-localization. The authors propose a framework to build smaller, faster models using width and resolution multipliers, improving size and speed with minimal accuracy loss. The paper concludes by mentioning plans to release MobileNet models in TensorFlow for wider adoption.

Karen Simonyan and Andrew Zisserman [26] explore the impact of convolutional network depth on accuracy in large-scale image recognition. The authors evaluate the performance of deep networks using very small (3×3) convolution filters, showing that increasing depth to 16-19 layers significantly improves classification accuracy. These findings were key to their success in the 2014 ImageNet Challenge, where they secured first and second places in the classification and localization tracks, respectively. The paper demonstrates that deeper networks generalize well to other datasets, achieving state-of-the-art results without the need for more complex models. By focusing on depth,

they show that ConvNets can be optimized for better performance without increasing computational cost. The authors have made their best-performing models publicly available to aid further research in deep visual representations.

2.2 SUMMARY OF CHALLENGES TO BE ADDRESSED

Several challenges remain in integrating AI with Ayurveda for Dosha identification. Existing systems often analyze individual features, such as skin or hair, leading to incomplete and inconsistent predictions. A holistic approach that incorporates multiple features, such as eyes, lips, and face shape, is needed for greater accuracy. Current datasets lack diversity in age, ethnicity, and regional variations, limiting scalability and robustness. Overfitting and poor generalization are common issues, and the lack of multi-modal analysis combining image and text data hinders real-time assessments. Additionally, scalability for low-resource devices, data privacy concerns, and limited collaboration with Ayurvedic practitioners further complicate adoption.

The proposed system addresses these limitations by adopting a holistic, multi-feature approach to Dosha identification. It analyzes various body characteristics—skin, hair, eyes, lips, and face shape—ensuring accurate and consistent classification even when individual features vary. The system tackles dataset challenges by utilizing advanced data augmentation techniques to create a diverse and balanced dataset, enhancing model performance. Transfer learning techniques reduce overfitting and improve accuracy, while scalable AI models ensure efficient deployment on low-resource devices. Collaboration with Ayurvedic practitioners bridges the gap between traditional practices and modern technology, ensuring alignment with holistic healthcare principles.

CHAPTER 3

SYSTEM ARCHITECTURE

Figure 3.1 illustrates the system architecture for Ayurveda dosha classification, which consists of three main stages: data collection, data augmentation and preprocessing, and model evaluation. The dataset includes images representing three hair types—straight, wavy, and curly—each associated with one of the Ayurvedic dosha types: Vata, Pitta, and Kapha. The images undergo augmentation and preprocessing steps, including resizing and normalization, to enhance model performance. Seven transfer learning models—VGG16, VGG19, ResNet50, MobileNet, InceptionV3, DenseNet121, and EfficientNet—are fine-tuned for dosha classification. These models are evaluated based on key performance metrics such as accuracy, precision, recall, and F1 score to determine the best model for deployment. The models are tested across five distinct datasets, with each dataset containing images classified by dosha type, ensuring a thorough evaluation of their performance. Figure 3.1 outlines the architecture of the Dosha classification system, which consists of four key modules:

Data Collection: This module includes web scraping and label annotation to gather and categorize data for the classification task. **Data Preprocessing:** It involves resizing images, performing data augmentation, and normalizing the images to prepare them for model training. **Transfer Learning Models:** Seven pre-trained transfer learning models—VGG16, VGG19, ResNet50, MobileNet, InceptionV3, DenseNet121, and EfficientNet—are fine-tuned for the Dosha classification task. **Evaluation:** The performance of the models is evaluated based on metrics such as accuracy, loss, precision, recall, F1 score, and the best model is selected for deployment.

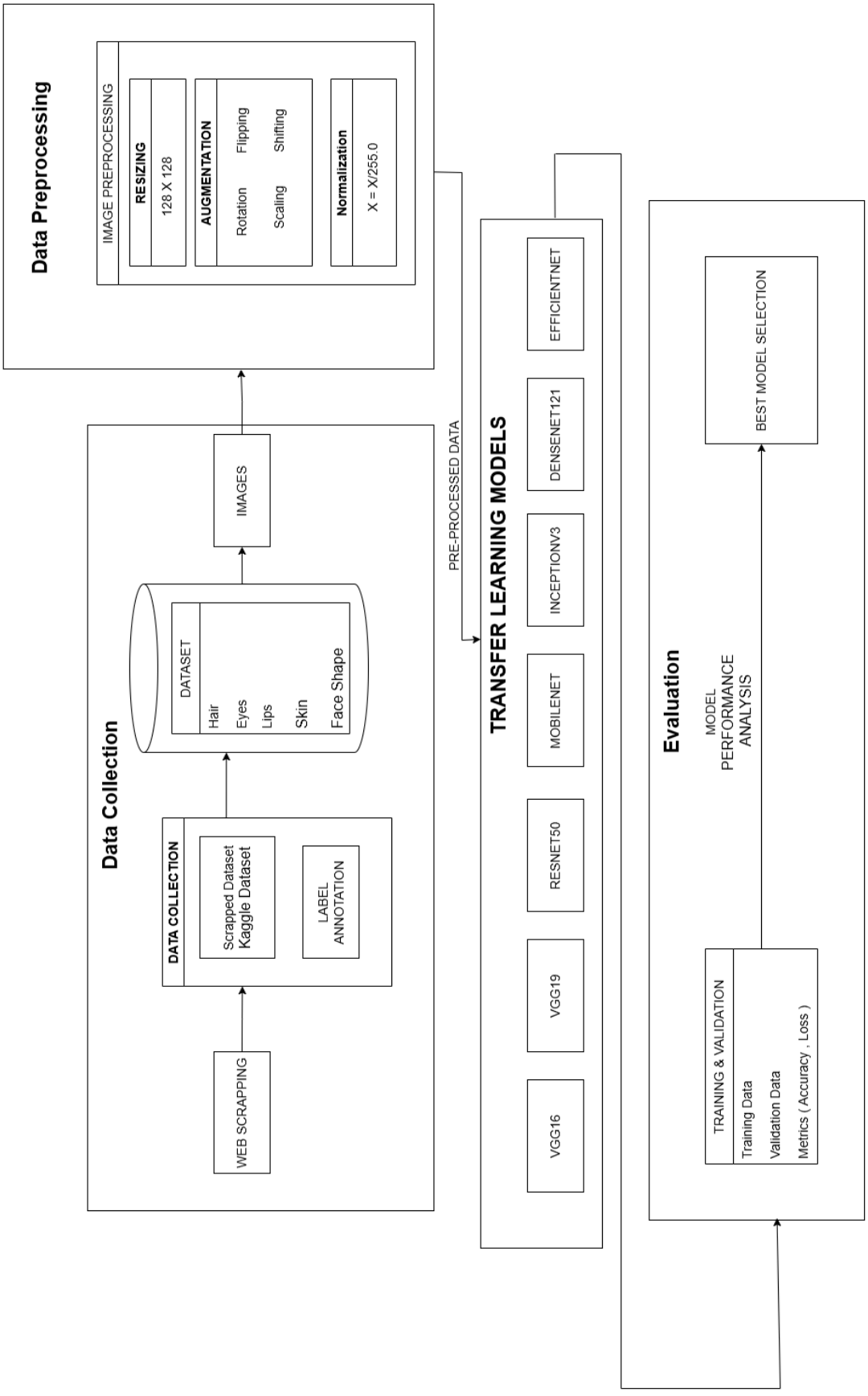


Figure 3.1: System Architecture of Dosha Classification

3.1 DATASET COLLECTION

The dataset was meticulously curated to enable Ayurvedic Dosha classification by incorporating images that represent distinct traits of hair, eyes, lips, skin types, and face shapes/chins. Figure 3.2 shows the details of the datasets used in this study, with images of straight, curly, and wavy hair types mapped to their corresponding Ayurvedic Dosha types: Vata (straight hair), Pitta (curly hair), and Kapha (wavy hair). The initial dataset was sourced from Kaggle, and to enhance its diversity and improve the model's robustness, additional images were manually added. The dataset was then split into training, validation, and test sets to facilitate comprehensive evaluation of the model's performance across different stages of training. The figure illustrates how these datasets are structured and their mapping to the respective Dosha types for effective classification.

Dataset	Total Images (Unaugmented)	Total Images (Augmented)	Classes (Original)	Classes (Mapped to Doshas)	Images per Class (Unaugmented)	Images per Class (Augmented)
Skin Dataset	76	650	dry, normal, oily	Vata (Dry), Kapha (Normal), Pitta (Oily)	dry: 22, normal: 24, oily: 30	dry: 186, normal: 203, oily: 261
Eyes Dataset	125	592	kapha, pittha, vata	Kapha (Blue smooth), Pitta (Gray/Yellow/Green), Vata (Black/Brown)	kapha: 32, pittha: 50, vata: 43	kapha: 197, pittha: 198, vata: 197
Lips Dataset	204	597	dry_cracked, inflammed, pale	Vata (Dry Cracked), Pitta (Inflamed), Kapha (Pale)	dry_cracked: 79, inflammed: 72, pale: 53	dry_cracked: 198, inflammed: 199, pale: 200
Face Shape Dataset	117	1050	Rounded_Double_Chin, Tapering_Triangular, Thin_Angular	Kapha (Rounded Double Chin), Pitta (Tapering Triangular), Vata (Thin Angular)	Rounded_Double_Chin: 9, Tapering_Triangular: 77, Thin_Angular: 31	Rounded_Double_Chin: 350, Tapering_Triangular: 350, Thin_Angular: 350
Hair Dataset	809	N/A	curly, straight, wavy	Kapha (Curly), Pitta (Straight), Vata (Wavy)	curly: 280, straight: 263, wavy: 266	N/A

Figure 3.2: Dataset details with features

For the eyes dataset, black and brown eyes were linked to Kapha, gray, yellowish, or green eyes to Pitta, and blue eyes with a smooth appearance to Vata. Similarly, the lips dataset categorized dry and cracked lips as Vata, inflamed or reddish lips as Pitta, and pale or whitish lips as Kapha. The skin

dataset further complemented this classification, associating dry skin with Vata, oily skin with Pitta, and normal skin with Kapha. In addition, a face shape and chin dataset was incorporated: thin, angular chins were linked to Vata; tapering or triangular chins to Pitta; and double chins to Kapha.

This enriched dataset, encompassing features from hair, eyes, lips, skin, and face shapes/chins, was carefully partitioned to ensure balanced representation for training and evaluation. By integrating diverse and holistic features, this dataset provides a strong foundation for accurate Dosha classification, bridging traditional Ayurvedic diagnostics with advanced deep learning models.

3.2 WEB SCRAPPING

In order to gather images for the dataset, a web scraping process was implemented using Selenium and BeautifulSoup. The script was designed to initialize the Selenium WebDriver, which was configured to use ChromeDriver, automatically installed via webdriver manager. It then navigated to the Google Image Search results page using the query “dry cracked lips images.” To ensure that more images were loaded, the page was scrolled three times, with a 2-second delay between each scroll to allow the images to load properly.The HTML content of the page was extracted using BeautifulSoup, which was then parsed to find all the image elements. Specifically, the ¡img¿ tags were located, and the src attribute, containing the image URLs, was retrieved. A directory named “scraped images” was created to store the downloaded images. The script then looped through the list of image URLs, downloading the content using the requests library and saving the images locally.The process was capped at downloading 150 images or when the available image URLs were exhausted. This approach ensured efficient collection of image data for further analysis

3.3 LABEL ANNOTATION

Labeling the images in this project is a critical step, especially because the datasets have been scraped and lack predefined labels. In order to classify these images according to Ayurvedic conditions, it is necessary to manually label the dataset according to specific categories such as skin type, face shape, hair type, eyes, and lips. This process allows the model to understand and differentiate between these characteristics based on Ayurvedic principles, which would otherwise be difficult with an unlabelled dataset. Using traditional labeling tools like LabelImg would not suffice due to the complexity of these categories, and the difficulty of manually categorizing these subtle distinctions in the images.

To address these challenges, I used V7 Labs, which provides an efficient and more accurate labeling system. This system allows me to easily label and categorize the five distinct datasets that are essential for Ayurvedic classification: Eyes, Lips, Hair, Skin, and Face Shape. With proper labeling, the model can learn the specific features of each category in alignment with Ayurvedic conditions, which significantly improves the training process and the quality of predictions. This structured approach ensures flexibility and consistency across the datasets and accelerates the overall development of the classification system.

1. **Create a New Project:** Log in to V7 Labs and initiate a new project. Choose ”Image Classification” as the annotation type. Upload your images either directly or from cloud storage.
2. **Define Categories:** Set up the categories (labels) that correspond to your images (e.g., Oily, Dry, Normal for skin types). Establish labeling guidelines to maintain consistency (e.g., defining the key features of each category).

3. **Annotate Images:** Open each image in the annotation interface. Assign the correct label to each image by selecting from the predefined labels. For multi-class classification, you can annotate an image with multiple categories. Optionally, use AI-assisted labeling to suggest labels based on pre-trained models.

4. **Review and Edit Annotations:** Double-check all images to ensure they are labeled correctly. If any mistakes are found, make the necessary updates to the annotations.

5. **Export Annotations:** Once the labeling process is complete, export the annotations in your preferred format (e.g., CSV, JSON, TXT). Download the dataset, which now includes both the images and their corresponding labels.

.

3.4 DATA AUGMENTATION AND PREPROCESSING

To load and preprocess the images, a function was defined to read all images from a specified folder, resize them to a uniform size of 128x128 pixels, and associate each image with a corresponding label. This function was then applied to load images for the three Dosha types: Kapha, Pitta, and Vata, assigning labels of 0, 1, and 2, respectively. To enhance the diversity of the dataset and improve model performance, data augmentation techniques were implemented. Augmentation included random cropping, where a portion of the image was randomly cropped and resized back to its original dimensions, scaling, where images were resized by a factor of 1.2 and then adjusted back to the target dimensions, and random perspective transformation, achieved by slightly shifting the four corners of the image to create a perspective effect. Each augmented image was paired with its corresponding label to maintain

consistency in the dataset. These steps ensured the creation of a richer and more robust dataset for training the model.

3.5 MODULES DESCRIPTION

Seven prominent transfer learning models were explored and compared for their performance in image classification tasks, particularly in medical and skin classification applications. Transfer learning was employed by utilizing pre-trained models, which had been trained on large-scale datasets like ImageNet. By leveraging the pre-trained weights, the models were able to utilize learned features from these vast datasets, leading to improved performance on smaller, domain-specific datasets. This approach proved beneficial in handling limited training data, as it mitigated overfitting and sped up the training process, making it an effective solution for the task at hand.

3.5.1 VGG16 Architecture

Figure 3.3 shows the architecture of VGG16 where it takes images of size $224 \times 224 \times 3$. It then passes through several convolutional layers organized into blocks, each consisting of convolutional layers followed by max-pooling. Block 1 uses 64 filters, Block 2 uses 128, Block 3 uses 256, and Blocks 4 and 5 use 512 filters, each with a 3×3 kernel and ReLU activation, followed by max-pooling with a 2×2 stride.

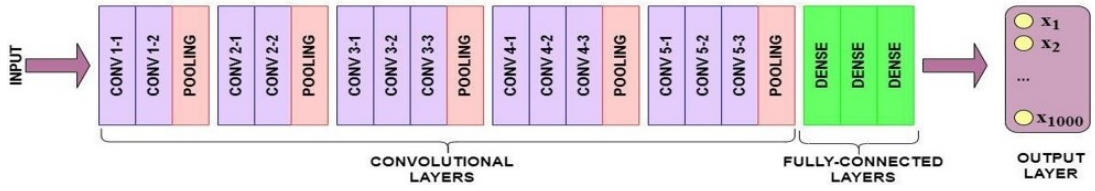


Figure 3.3: VGG16 architecture diagram

After the convolutional blocks, the output is passed through two fully connected layers, each with 4096 neurons and ReLU activation. The final output layer has one neuron with either softmax (for multi-class classification) or sigmoid (for binary classification) activation. Dropout regularization is applied to the fully connected layers to prevent overfitting.

3.5.2 RESNET-50 Architecture

Figure 3.4 shows the architecture of RESNET50 where it takes images of size $224 \times 224 \times 3$. The model starts with an initial convolutional layer using 64 filters of size 7×7 and a stride of 2, followed by Batch Normalization and ReLU activation. A max-pooling layer of size 3×3 with stride 2 follows. The core of ResNet-50 consists of four blocks of residual units: Block 1 with 3 units, Block 2 with 4 units, Block 3 with 6 units, and Block 4 with 3 units. These blocks use filters of increasing size, from 64 to 512, and allow for deeper networks through residual connections that help to combat the vanishing gradient problem.

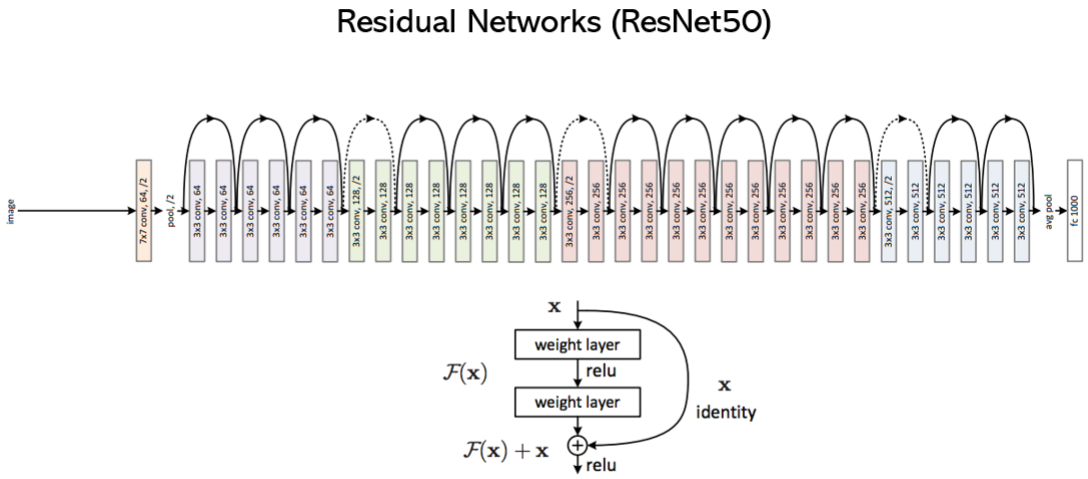


Figure 3.4: ResNet50 architecture diagram

The architecture utilizes a bottleneck design for each residual unit, which includes 1×1 convolutions for dimension reduction and restoration, as well as 3×3 convolutions for feature extraction. After the residual blocks, global average pooling reduces the feature maps to a $1 \times 1 \times 2048$ tensor. The output layer consists of a fully connected layer with one neuron, applying softmax or sigmoid activation depending on the classification type. Dropout regularization is applied to the fully connected layers to reduce overfitting and improve the model's generalization performance.

3.5.3 VGG19 Architecture

Figure 3.5 shows the architecture of VGG19 where it takes images of size $224 \times 224 \times 3$ (height, width, channels). It consists of five convolutional blocks. Block 1 has two convolutional layers with 64 filters of size 3×3 and ReLU activation, followed by a max-pooling layer with a 2×2 filter. Block 2 follows a similar structure with 128 filters. Blocks 3, 4, and 5 each contain four convolutional layers with increasing filter sizes: 256 and 512, all using ReLU activation followed by max-pooling layers with a 2×2 filter. These layers enable the model to extract hierarchical features from the input image at different spatial resolutions.

Figure 3.5: VGG19 architecture diagram

After the convolutional layers, the model transitions into fully connected layers. The first two fully connected layers consist of 4096 neurons

each, with ReLU activation to model non-linear relationships. The final output layer has a single neuron, and depending on the task, it uses either softmax activation for multi-class classification or sigmoid activation for binary classification. To prevent overfitting, dropout regularization is applied to the fully connected layers, ensuring better generalization during training.

3.5.4 MOBILENET Architecture

Figure 3.6 shows the architecture of MOBILENET where it takes images of size $224 \times 224 \times 3$ (height, width, channels). It employs depthwise separable convolutions to reduce computational complexity. In these convolutions, each block consists of a depthwise convolution, which applies a single filter per input channel for spatial filtering, and a pointwise convolution, which applies a 1×1 convolution to combine features across channels. The architecture includes an initial convolution with 32 filters of size 3×3, stride 2, followed by batch normalization and ReLU6 activation. Subsequently, several depthwise separable blocks are used: Block 1 with 64 filters and stride 1, Block 2 with 128 filters and stride 2, and others with varying filter sizes and strides. Blocks 7-12 are repeated with 512 filters and stride 1, and the final block uses 1024 filters with stride 1.

After the convolutional layers, global average pooling is applied to reduce the spatial dimensions to a $1 \times 1 \times 1024$ tensor. The output layer consists of a fully connected layer with one neuron and uses softmax for multi-class classification or sigmoid for binary classification. ReLU6 activation is used throughout the network to enhance performance on mobile devices. To prevent overfitting, dropout regularization is applied before the fully connected layer. This approach makes MobileNet an efficient and effective model for mobile and resource-constrained environments.

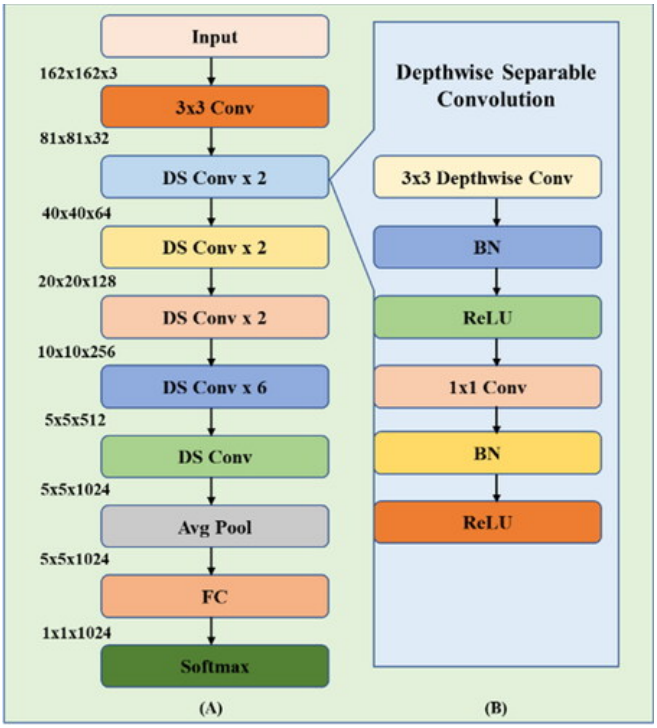


Figure 3.6: MobileNet architecture diagram

3.5.5 INCEPTIONV3 Architecture

Figure 3.7 shows the architecture of INCEPTIONV3 where it takes takes input images of size $299 \times 299 \times 3$. The initial layers consist of convolution with 32 filters of size 3×3, stride 2, followed by Batch Normalization and ReLU. This is followed by another convolution with 32 filters and a final convolution with 64 filters of size 3×3 with padding. MaxPooling with a size of 3×3 and stride 2 is applied. InceptionV3 uses several Inception modules, including Module A with parallel convolutions of 1×1, 3×3, and 5×5 filters, Module B with 1×1 convolutions followed by asymmetric convolutions, and Module C with factorized 1×1 and 3×3 convolutions.

The model includes two auxiliary classifiers to prevent vanishing gradients, each with fully connected layers and softmax activation. Reduction

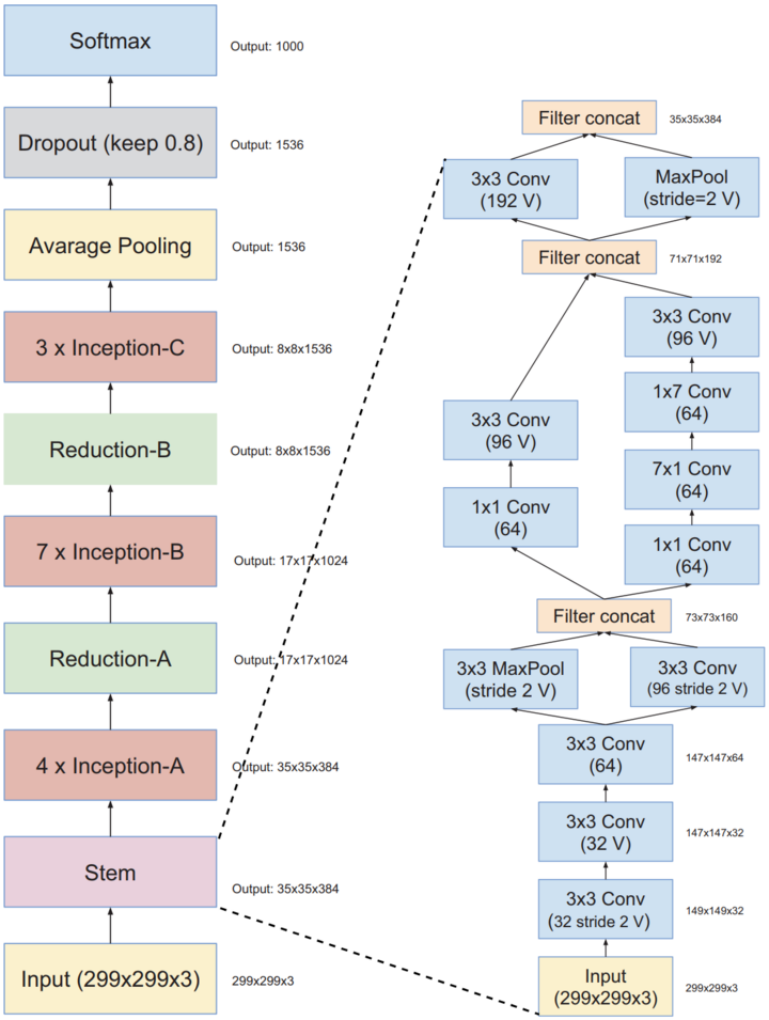


Figure 3.7: InceptionV3 architecture diagram

modules are used for downsampling. After convolutional layers, global average pooling reduces the dimensions to $1 \times 1 \times 2048$. The final output is a fully connected layer with one neuron and softmax or sigmoid activation. Dropout is applied before the fully connected layer to prevent overfitting. InceptionV3 also replaces larger convolutions with smaller factorized convolutions for improved efficiency.

3.5.6 DENSENET121 Architecture

Figure 3.8 shows the architecture of DENSENET121 where it takes

input images of size $224 \times 224 \times 3$. The initial layers include a convolution with 7×7 filters, stride 2, followed by Batch Normalization and ReLU activation. A MaxPooling layer with size 3×3 and stride 2 follows. DenseNet121 consists of four dense blocks, where each block contains densely connected convolutional layers. The first block has 6 layers, the second block has 12 layers, the third block has 24 layers, and the final block contains 16 layers. Transition layers between blocks reduce the feature map depth using 1×1 convolutions followed by average pooling with size 2×2 and stride 2.

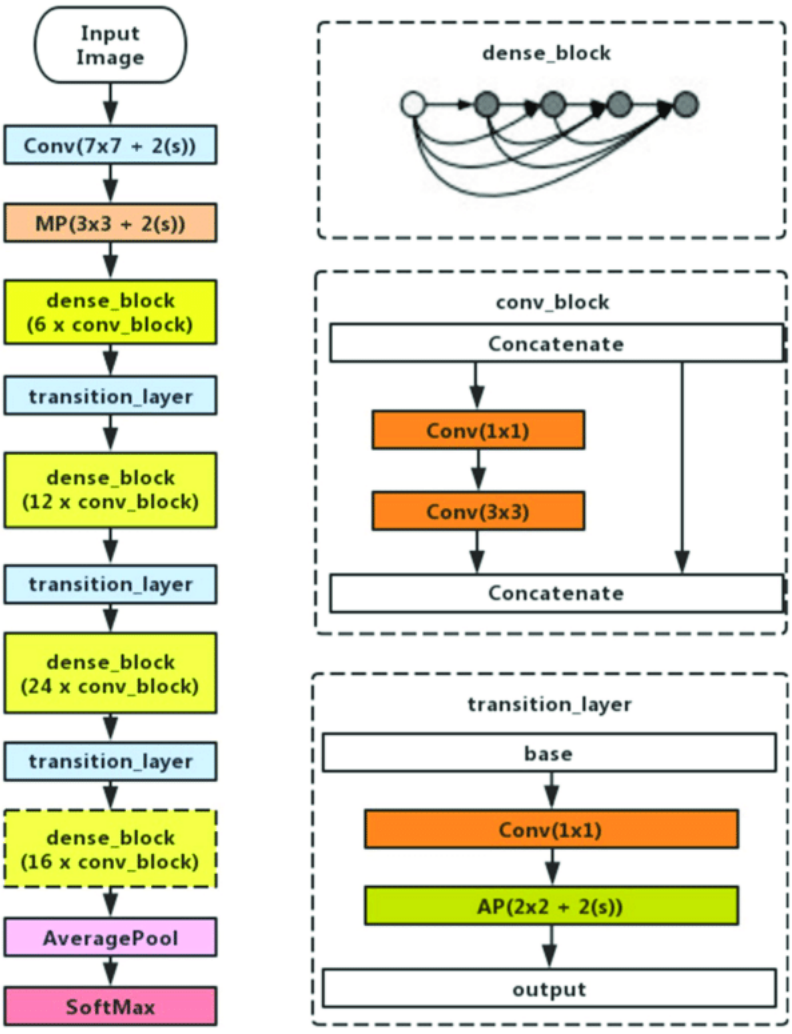


Figure 3.8: DenseNet121 architecture diagram

After the final dense block, a global average pooling layer reduces the spatial dimensions to $1 \times 1 \times 1024$. The fully connected output layer has one neuron with softmax or sigmoid activation. Dropout regularization is applied before the fully connected layer to prevent overfitting. DenseNet121's dense connections ensure each layer receives input from all previous layers, improving gradient flow and mitigating the vanishing gradient problem.

3.5.7 EFFICIENTNET Architecture

Figure 3.9 shows the architecture of EFFICIENTNET where it takes input layer that accepts images of size $224 \times 224 \times 3$. The architecture starts with an initial convolution using 3×3 filters and stride 2, followed by Batch Normalization and Swish activation. EfficientNet employs a compound scaling method, simultaneously scaling the depth, width, and resolution of the network. This is achieved through the use of Mobile Inverted Bottleneck Convolutions (MBConv), which consist of depthwise separable convolutions followed by 1×1 convolutions to reduce computational complexity. Additionally, squeeze-and-excitation layers are applied after each MBConv block to recalibrate the channel-wise feature responses. The compound scaling technique balances the trade-off between model size, computational cost, and performance.

After the final MBConv block, a global average pooling layer reduces the spatial dimensions to $1 \times 1 \times 1280$ (depending on the model variant). The output is then passed through a fully connected layer with one neuron, utilizing either softmax for multi-class classification or sigmoid for binary classification. Dropout regularization is applied before the fully connected layer to prevent overfitting. The Swish activation function is used throughout the network in place of ReLU, providing smoother gradients and enhancing performance, especially in deeper networks.

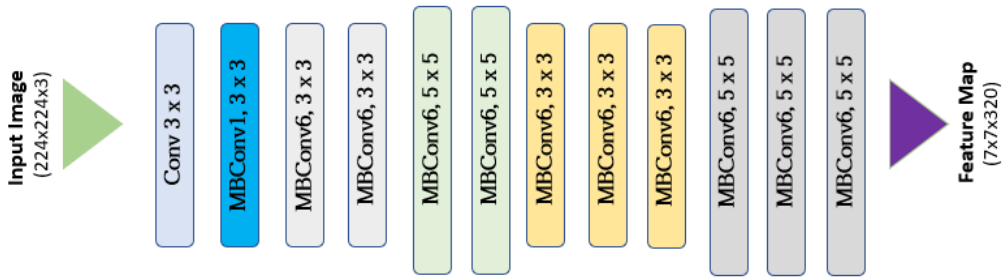


Figure 3.9: EfficientNet architecture diagram

3.6 ARCHITECTURE FLOW

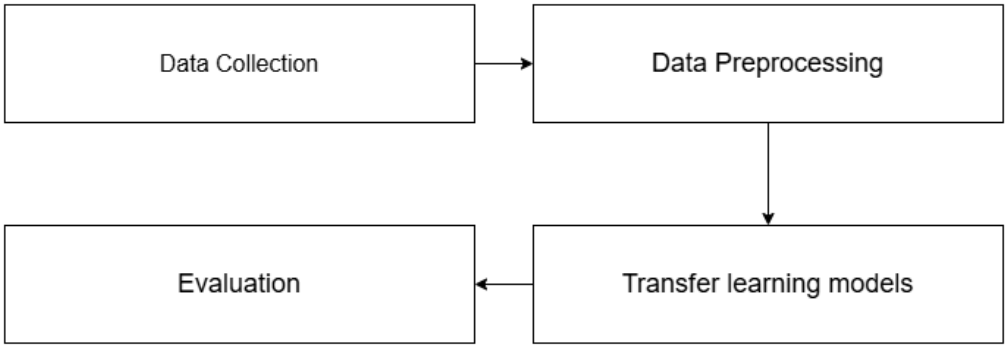


Figure 3.10: Architecture flow diagram

The model training process begins with data collection, where the dataset is split into training, validation, and test sets. Preprocessing includes resizing, normalizing, and applying data augmentation techniques like rotation and flipping to increase dataset diversity. The data is then fed into a pre-trained model for transfer learning, where fine-tuning is done by freezing early layers and adding custom layers. The model is compiled with a suitable loss function and optimizer, and training occurs over multiple epochs using mini-batch gradient descent.

During training, the model's performance is evaluated on the validation set using metrics like accuracy, precision, recall, and F1 score. Techniques like early stopping prevent overfitting, while learning rate adjustments help optimize training. Visualization of training curves offers insight into the model's progress. Hyperparameters are tuned to refine the model's performance, ensuring it generalizes well to unseen data.

Once trained, the model is evaluated on the test set, providing an unbiased estimate of its performance. Confusion matrices and evaluation metrics help identify areas for improvement. If performance is satisfactory, the model is ready for deployment; otherwise, adjustments to the architecture or hyperparameters are made. The final test set evaluation confirms the model's ability to generalize, providing a final check before deployment.

3.7 BEST MODEL SELECTION

In order to determine the optimal model for Ayurvedic dosha classification, the performance of six transfer learning models—VGG16, VGG19, ResNet50, MobileNet, InceptionV3, Efficientnet, and DenseNet121—was compared in this project. The capacity of each model to categorize the three doshas—Pitta, Kapha, and Vata—based on hair types (straight, wavy, and curly) was assessed using important performance measures like accuracy, precision, recall, and F1 score.

Model Selection and Pre-processing: Pre-trained models (VGG16, VGG19, ResNet50, MobileNet, InceptionV3, EfficientNet, DenseNet121) were fine-tuned for the dosha classification task. Image preprocessing included resizing, normalization, and data augmentation to improve generalization.

Hyperparameter Tuning: Hyperparameters such as learning rate, batch size, and epochs were optimized using grid or random search techniques to find the best model settings.

Model Training and Fine-tuning: Pre-trained models were initially frozen and gradually fine-tuned with new layers for dosha classification. The top layers were unfrozen to adapt the models to the new dataset.

Evaluation Metrics: Performance was measured using accuracy, precision, recall, and F1 score to assess classification effectiveness for the three doshas.

Model Comparison: MobileNet performed the best for efficiency and accuracy, followed by DenseNet121 and InceptionV3. VGG16 and VGG19 were computationally efficient but less effective. ResNet50 and EfficientNet underperformed compared to the others.

The best model for Dosha classification will be selected based on performance metrics such as accuracy, precision, recall, and F1 score. The final model selection will prioritize the one that offers the best balance between computational efficiency and classification accuracy, ensuring optimal performance for real-world deployment. The selected model will be fine-tuned further to ensure robust performance across diverse datasets.

Moreover, attention will be given to minimizing overfitting and enhancing the model's generalization capabilities. Techniques such as cross-validation and hyperparameter tuning will be employed to achieve optimal results. Ultimately, the goal is to create a reliable, scalable, and efficient AI-driven solution that aligns with Ayurvedic diagnostic principles.

CHAPTER 4

IMPLEMENTATION DETAILS

The implementation chapter focuses on applying transfer learning models for Dosha classification, merging Ayurvedic principles with advanced AI techniques to achieve accurate identification.

4.1 DATASETS FOR DOSHA CLASSIFICATION

4.1.1 Hair Dataset

The hair dataset was sourced from Kaggle, consisting of images of three distinct hair types: straight, curly, and wavy. Each hair type was mapped to an Ayurvedic Dosha as detailed in Table 4.1:

Table 4.1: Hair Dataset Summary

Attribute	**Value**
Total Images (Unaugmented)	809
Total Images (Augmented)	N/A
Classes (Original)	curly, straight, wavy
Classes (Mapped to Doshas)	Kapha (Curly), Pitta (Straight), Vata (Wavy)
Images per Class (Unaugmented)	curly: 280, straight: 263, wavy: 266
Images per Class (Augmented)	N/A

To increase diversity and improve model generalization, additional images were manually curated and added to the dataset. The augmented dataset was split into training, validation, and test sets to ensure balanced representation and robust model evaluation.

4.1.2 Eyes Dataset

Images representing various eye colors and textures were collected to build the eyes dataset. Each eye characteristic was mapped to the corresponding Dosha based on Ayurvedic principles as shown in Table 4.2:

Table 4.2: Eyes Dataset Summary

Attribute	**Value**
Total Images (Unaugmented)	125
Total Images (Augmented)	592
Classes (Original)	kapha, pitta, vata
Classes (Mapped to Doshas)	(Blue smooth),(Gray/Yellow/Green), (Black/Brown)
Images per Class (Unaugmented)	kapha: 32, pitta: 50, vata: 43
Images per Class (Augmented)	kapha: 197, pitta: 198, vata: 197

4.1.3 Lips Dataset

The lips dataset included images showcasing distinct physical attributes associated with each Dosha as detailed in Table 4.3:

Table 4.3: Lips Dataset Summary

Attribute	**Value**
Total Images (Unaugmented)	204
Total Images (Augmented)	597
Classes (Original)	dry_cracked, inflamed, pale
Classes (Mapped to Doshas)	Vata (Dry Cracked), Pitta (Inflamed), Kapha (Pale)
Images per Class (Unaugmented)	dry_cracked: 79, inflamed: 72, pale: 53
Images per Class (Augmented)	dry_cracked: 198, inflamed: 199, pale: 200

Images were labeled according to these characteristics to create a comprehensive dataset for the model.

4.1.4 Skin Dataset

The skin dataset was categorized into three types, each linked to a Dosha. This categorization reflects Ayurvedic principles of physical traits corresponding to the Doshas as detailed in Table 4.4: The dataset was

Table 4.4: Skin Dataset Summary

Attribute	**Value**
Total Images (Unaugmented)	76
Total Images (Augmented)	650
Classes (Original)	dry, normal, oily
Classes (Mapped to Doshas)	Vata (Dry), Kapha (Normal), Pitta (Oily)
Images per Class (Unaugmented)	dry: 22, normal: 24, oily: 30
Images per Class (Augmented)	dry: 186, normal: 203, oily: 261

preprocessed, balanced, and split into training, validation, and test sets to ensure robust performance of the classification models.

4.1.5 Face Shape Dataset

The face shape dataset comprised images categorized into three distinct facial structures, each linked to a Dosha and shown in Table 4.5:

classes :Rounded Double Chin, Tapering Triangular, Thin Angular

Table 4.5: Face Shape Dataset Summary

Attribute	**Value**
Total Images (Unaugmented)	117
Total Images (Augmented)	1050
Classes (Mapped to Doshas)	Kapha (R_D_C), Pitta (T_T), Vata (T_A)
Images per Class (Unaugmented)	R_D_C: 9, T_T: 77, T_: 31
Images per Class (Augmented)	R_D_C: 350, T_T: 350, T_A: 350

These facial features were carefully labeled to align with Ayurvedic principles. This dataset adds a unique dimension to Dosha classification by incorporating facial morphology as a diagnostic factor. Similar to the

other datasets, the face shape dataset was preprocessed, balanced, and split into training, validation, and test sets to ensure consistent and reliable model performance.

4.1.6 Overall Dataset Details

By building separate datasets for hair, eyes, lips, skin, and face shapes, the classification process aligns with Ayurvedic diagnostic principles. Each dataset independently maps distinct physical traits to their corresponding Doshas, ensuring a focused and detailed analysis for each domain. This approach allows the models to specialize in understanding Ayurvedic principles specific to each dataset while maintaining robust accuracy in Dosha classification.

Dataset	Total Images (Unaugmented)	Total Images (Augmented)	Classes (Original)	Classes (Mapped to Doshas)	Images per Class (Unaugmented)	Images per Class (Augmented)
Skin Dataset	76	650	dry, normal, oily	Vata (Dry), Kapha (Normal), Pitta (Oily)	dry: 22, normal: 24, oily: 30	dry: 186, normal: 203, oily: 261
Eyes Dataset	125	592	kapha, pittha, vata	Kapha (Blue smooth), Pitta (Gray/Yellow/Green), Vata (Black/Brown)	kapha: 32, pittha: 50, vata: 43	kapha: 197, pittha: 198, vata: 197
Lips Dataset	204	597	dry_cracked, inflammed, pale	Vata (Dry Cracked), Pitta (Inflamed), Kapha (Pale)	dry_cracked: 79, inflammed: 72, pale: 53	dry_cracked: 198, inflammed: 199, pale: 200
Face Shape Dataset	117	1050	Rounded_Double_Chin, Tapering_Triangular, Thin_Angular	Kapha (Rounded Double Chin), Pitta (Tapering Triangular), Vata (Thin Angular)	Rounded_Double_Chin: 9, Tapering_Triangular: 77, Thin_Angular: 31	Rounded_Double_Chin: 350, Tapering_Triangular: 350, Thin_Angular: 350
Hair Dataset	809	N/A	curly, straight, wavy	Kapha (Curly), Pitta (Straight), Vata (Wavy)	curly: 280, straight: 263, wavy: 266	N/A

Figure 4.1: Dataset details

Figure 4.1 provides an overview of the dataset details, including the distribution of images across different classes. It highlights the structure and organization of the dataset used for training and evaluating the models.

4.2 WEB SCRAPING

Algorithm 4.1, defines a process for scraping and downloading images. The `download_image` function saves an image from a valid URL to a specified folder, naming it as `image_count.jpg`, or prints an error if the download fails.

Algorithm 4.1 WEB SCRAPING

Input: `page_link` (URL of the webpage to scrape)
Output: Downloaded images saved in the specified folder

```
1:  function DOWNLOAD_IMAGE(url, folder, count)
2:      if GET(url) is successful then
3:          Save image as 'image_count.jpg'
4:      else
5:          Print error message
6:      end if
7:  end function
8:  function SCRAPE_IMAGES(url, folder, limit)
9:      if folder doesn't exist then
10:         Create folder
11:     end if
12:     response = GET(url)
13:     if response is successful then
14:         for each image in response do
15:             if count ≥ limit then
16:                 break
17:             end if
18:             Get image URL
19:             if URL is valid then
20:                 DOWNLOAD_IMAGE(url, folder, count)
21:             end if
22:             Increment count
23:         end for
24:     else
25:         Print error message
26:     end if
27: end function
28: SCRAPE_IMAGES(url, folder, limit=60)
```

The `scrape_images` function checks if the folder exists (creating it if necessary), fetches images from a URL, and iterates through them to download up to a specified limit. It validates each image URL before calling `download_image`. The function is then called with a URL, folder, and a limit of 60 images.

DOWNLOADED IMAGES

Table 4.6: Web scrapped and downloaded images

Figure 4.6 displays the web-scraped and downloaded images from each dataset. These images were collected from various sources to ensure diverse representation for each category in the project, aiding in the model's ability to generalize and perform accurately across different inputs.

4.3 LABEL ANNOTATION

Label annotation involved manually tagging images with the appropriate dosha types (Vata, Pitta, and Kapha) based on physical features such

as hair type, skin condition, and facial characteristics. This process ensured accurate dataset labeling for training deep learning models for Ayurvedic dosha classification.

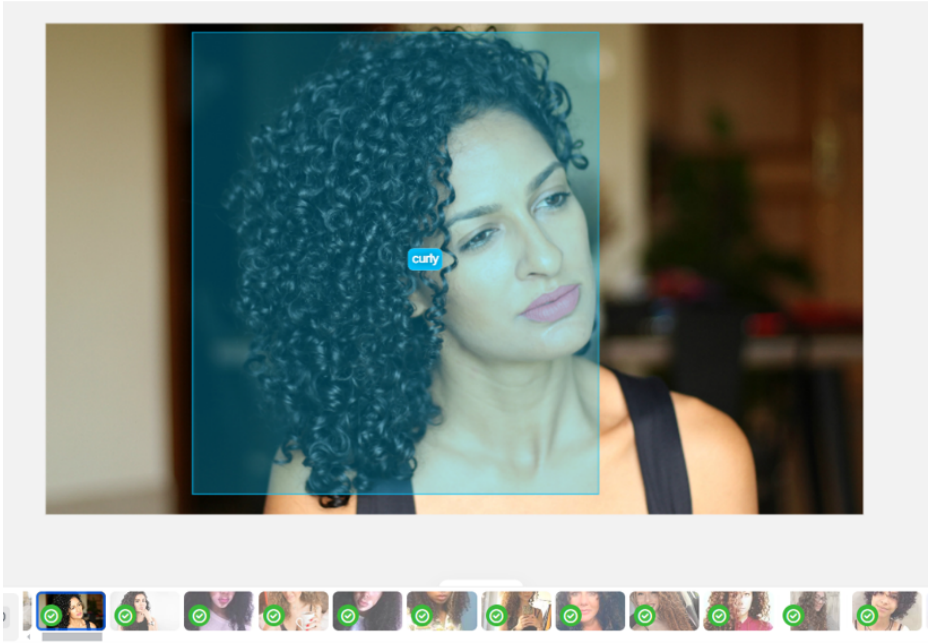


Figure 4.2: Label Annotation

Figure 4.2 represents how specific regions of the image, such as hair, skin, and facial features, are annotated using the V7 Labs tool. This process allows for precise labeling of the dosha-related attributes, enabling accurate model training.

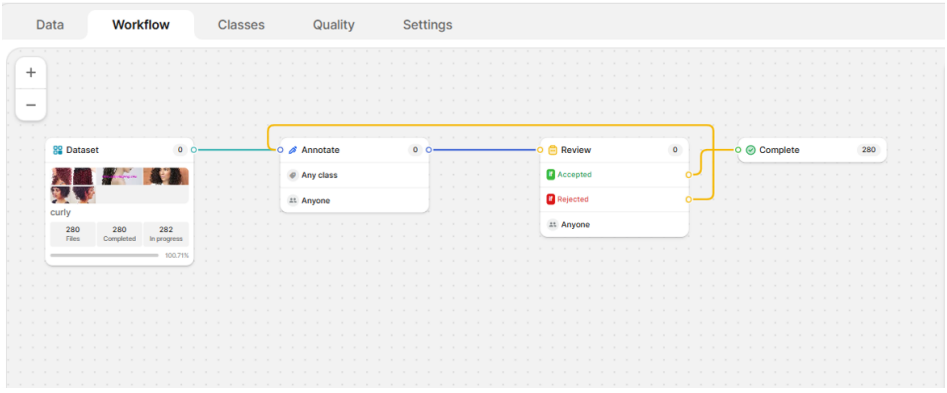


Figure 4.3: Label Annotation

Figure 4.3 illustrates the workflow of the V7 Labs tool for label annotation, highlighting the steps involved in annotating images with the appropriate dosha-related labels. This workflow ensures systematic and accurate labeling for model training.

4.4 PREPROCESS AND AUGMENTATION

4.4.1 Preprocessing

The images were loaded from their respective directories and resized to a fixed dimension of 128×128 pixels. This resizing ensures uniformity across all images, regardless of their original dimensions. The resizing step was applied to all images before any augmentation techniques were used, creating a consistent input format for subsequent processes.

The choice of resizing to 128×128 pixels was made to balance computational efficiency and the retention of sufficient image details. This resolution is compact enough to reduce computational load and memory usage during model training, while still being large enough to preserve essential features in the images. Using 128×128 ensures faster processing without significantly compromising data quality, making it an ideal choice for image classification tasks.Additionally, this standard size aligns well with the input requirements of many deep learning models, facilitating smooth integration with pre-trained architectures such as VGG16 or custom Convolutional Neural Networks (CNNs).

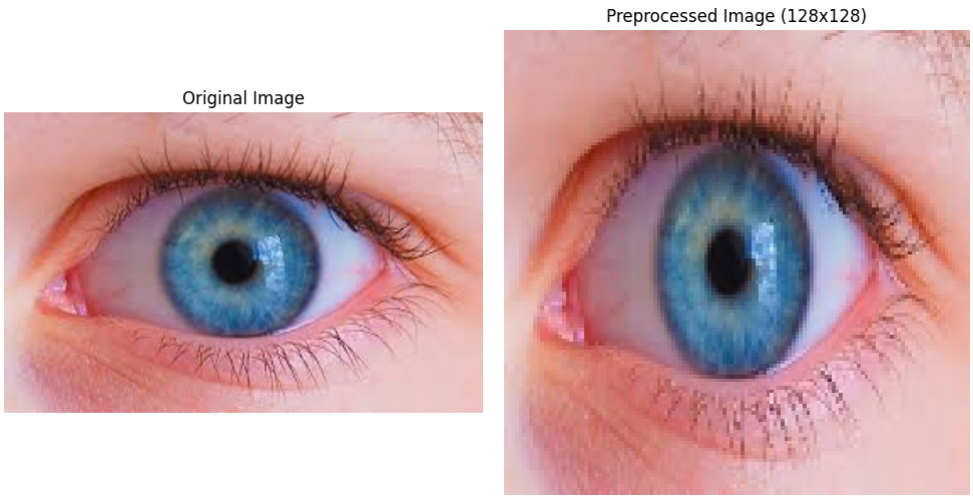


Figure 4.4: Preprocessed Image

Figure 4.4 displays the preprocessing of images, where the original images are resized to 128×128 dimensions. This step ensures uniformity in input size for the model training process.

4.4.2 Data Augmentation

Data Augmentation (Algorithm 4.2) was applied to the preprocessed images to artificially increase the size of the dataset and enhance the model's ability to generalize. Various augmentation techniques were utilized, including random cropping, where a random region of the image was selected and resized back to 128×128 pixels; scaling, which involved rescaling the image dimensions by a random factor (e.g., $1.2\times$) and resizing it back to 128×128 pixels; and perspective transformations, which introduced random distortions to simulate different perspectives.

Algorithm 4.2 Image Augmentation Function

Input: Normal image

Output: Augmented image (with random cropping, scaling, and perspective transformation)

function AUGMENT_IMAGES(images, label)
    Initialize augmented_images as an empty list
    for each image `img, _` in `images` **do**
        Get the height and width: (h, w) of `img`
        # Random Cropping
        $x\_start \leftarrow$ Random integer from 0 to $\lfloor 0.1 \cdot w \rfloor$
        $y\_start \leftarrow$ Random integer from 0 to $\lfloor 0.1 \cdot h \rfloor$
        cropped_img ← img[y_start:y_start + h - 20, x_start:x_start + w - 20]
        resized_cropped_img ← Resize cropped_img to (w, h)
        Append (resized_cropped_img, label) to augmented_images
        # Scaling
        scaled_img ← Resize img by scaling factor 1.2 along both axes
        resized_scaled_img ← Resize scaled_img to (w, h)
        Append (resized_scaled_img, label) to augmented_images
        # Random Perspective Transformation
        Define original points $pts1 = \{(0,0),(w-1,0),(0,h-1),(w-1,h-1)\}$
        Define transformed points $pts2$ by randomly adjusting coordinates within bounds
        matrix ← Compute perspective transformation matrix using $pts1$ and $pts2$
        perspective_img ← Apply perspective transformation to img using matrix
        Append (perspective_img, label) to augmented_images
    end for
    return augmented_images
end function

These transformations created diverse variations of the original images, thereby enriching the dataset and improving the robustness of the model. By incorporating these techniques, the model was exposed to a wide range of image variations, which improved its ability to handle real-world variability and reduced the risk of overfitting.

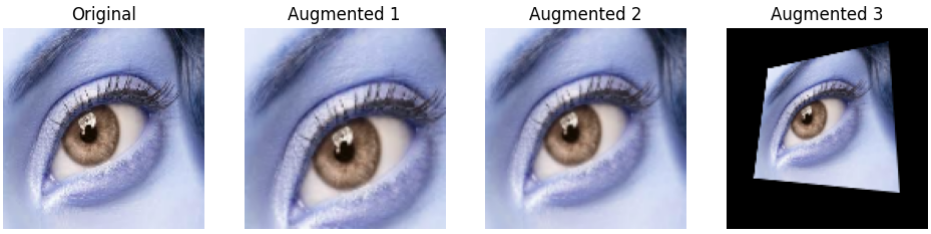


Figure 4.5: Augmented eye Image

Figure 4.5 displays the augmented images generated through various transformations, including random cropping, scaling, and perspective distortions. These augmented images enhance the dataset by introducing diverse variations, which improve the model's ability to generalize.

4.4.3 Normalization

After augmentation, pixel values of all images (both original and augmented) were scaled to a range of [0, 1] by dividing by 255. This normalization step ensured compatibility with the VGG-16 model, which requires input images with pixel values in this range. Additionally, the images were resized to 128x128 pixels to match the input size expected by VGG-16. The dataset was then split into training and testing sets to evaluate the model's performance effectively.

4.5 TRANSFER LEARNING MODELS

The seven transfer learning models used in this study include VGG16, VGG19, ResNet50, MobileNet, InceptionV3, DenseNet121, and Xception. These models leverage pre-trained weights to extract features from images, and by freezing the base layers, they are adapted to specific tasks with minimal data. Custom layers are added on top, allowing fine-tuning for improved performance and faster training. This approach effectively boosts

accuracy and generalization, even with limited data.

Algorithm 4.3 Model Training and Evaluation for Multiple Models

Input: Preprocessed image data (training and test datasets)
Output: Predicted labels (classification results)

for each model_name in model_list **do**
    Load the pre-trained base model (e.g., MOBILENET, VGG16, VGG19, RESNET50, INCEPTIONV3, EFFICEINTNET).
    Freeze the base model weights.
    Add custom layers to the base model:
        - GlobalAveragePooling2D
        - Dense layer (ReLU activation, L2 regularization)
        - Dropout layer (e.g., 0.5)
        - Final Dense layer (Softmax activation)
    Compile the model with:
        optimizer = Adam(learning_rate)
        loss = sparse_categorical_crossentropy
        metrics = [accuracy]
    Initialize EarlyStopping callback:
        early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
    Train the model using the training data:
        history = model.fit(training_data, labels, epochs=epochs, validation_split=validation_split, batch_size=batch_size, callbacks=[early_stopping])
    Save the trained model as `model_name_trained.h5`.
    Evaluate the model on the test data and generate a classification report:
        predictions = model.predict(test_data)
        Generate classification report (precision, recall, F1-score)
    Plot and display the training and validation accuracy history:
        Plot(history.history['accuracy'], history.history['val_accuracy'])
end for

Algorithm 4.3 outlines the process for training seven different pre-trained transfer learning models—MobileNet, VGG16, VGG19, ResNet50, InceptionV3, DenseNet121, and EfficientNetB0. These models are initialized with their base weights frozen, and custom layers are added, including a GlobalAveragePooling2D layer for dimensionality reduction and spatial feature aggregation, a Dense layer configured with ReLU activation and L2 regularization to enhance feature learning, a Dropout layer with a rate of

0.5 to prevent overfitting, and a Softmax output layer for classifying images into one of three categories: *Kapha*, *Pitta*, or *Vata*. The model is compiled using the Adam optimizer with a learning rate of 10^{-4}, sparse categorical cross-entropy as the loss function, and accuracy as the evaluation metric. The Skin Dataset, consisting of 128×128 RGB images, is used for this process, with 611 images (77.71%) allocated for training, 50 images (6.36%) for validation, and 175 images (22.24%) for testing. Early stopping is employed to monitor the validation loss and prevent overfitting. After training, the final model is saved and evaluated on the test set to assess classification performance using precision, recall, and F1-score. Training and validation accuracy plots are generated to analyze the model's learning progression. This approach ensures consistency and robustness in evaluating the models' ability to map skin types to doshas across the categories of Kapha,Pitta, and Vata.

```
Epoch 1/10
22/22 ━━━━━━━━━━━━━━━━━━━━ 0s 755ms/step - accuracy: 0.3816 - loss: 2.3835
Epoch 1: val_accuracy improved from -inf to 0.65714, saving model to best_model.keras
22/22 ━━━━━━━━━━━━━━━━━━━━ 38s 1s/step - accuracy: 0.3839 - loss: 2.3568 - val_accuracy: 0.6571 - val_loss: 0.8898
Epoch 2/10
22/22 ━━━━━━━━━━━━━━━━━━━━ 0s 807ms/step - accuracy: 0.6085 - loss: 0.8605
Epoch 2: val_accuracy improved from 0.65714 to 0.76000, saving model to best_model.keras
22/22 ━━━━━━━━━━━━━━━━━━━━ 22s 1s/step - accuracy: 0.6106 - loss: 0.8578 - val_accuracy: 0.7600 - val_loss: 0.6746
Epoch 3/10
22/22 ━━━━━━━━━━━━━━━━━━━━ 0s 754ms/step - accuracy: 0.7249 - loss: 0.6676
Epoch 3: val_accuracy improved from 0.76000 to 0.81143, saving model to best_model.keras
22/22 ━━━━━━━━━━━━━━━━━━━━ 21s 984ms/step - accuracy: 0.7265 - loss: 0.6654 - val_accuracy: 0.8114 - val_loss: 0.5458
Epoch 4/10
22/22 ━━━━━━━━━━━━━━━━━━━━ 0s 767ms/step - accuracy: 0.8257 - loss: 0.4938
Epoch 4: val_accuracy improved from 0.81143 to 0.84571, saving model to best_model.keras
22/22 ━━━━━━━━━━━━━━━━━━━━ 22s 988ms/step - accuracy: 0.8260 - loss: 0.4931 - val_accuracy: 0.8457 - val_loss: 0.4437
Epoch 5/10
22/22 ━━━━━━━━━━━━━━━━━━━━ 0s 873ms/step - accuracy: 0.8497 - loss: 0.4225
Epoch 5: val_accuracy did not improve from 0.84571
22/22 ━━━━━━━━━━━━━━━━━━━━ 23s 1s/step - accuracy: 0.8495 - loss: 0.4225 - val_accuracy: 0.8114 - val_loss: 0.4813
Epoch 6/10
22/22 ━━━━━━━━━━━━━━━━━━━━ 0s 817ms/step - accuracy: 0.8799 - loss: 0.3716
Epoch 6: val_accuracy did not improve from 0.84571
22/22 ━━━━━━━━━━━━━━━━━━━━ 23s 1s/step - accuracy: 0.8800 - loss: 0.3707 - val_accuracy: 0.8457 - val_loss: 0.3703
Epoch 7/10
...
22/22 ━━━━━━━━━━━━━━━━━━━━ 0s 749ms/step - accuracy: 0.9402 - loss: 0.1806
Epoch 10: val_accuracy improved from 0.92571 to 0.93714, saving model to best_model.keras
22/22 ━━━━━━━━━━━━━━━━━━━━ 21s 966ms/step - accuracy: 0.9404 - loss: 0.1805 - val_accuracy: 0.9371 - val_loss: 0.1937
6/6 ━━━━━━━━━━━━━━━━━━━━ 5s 772ms/step
```

Figure 4.6: Model Training

Figure 4.6 shows the training and validation accuracy progression for the seven transfer learning models. It highlights how each model adapts over epochs, helping to evaluate their performance and generalization ability. The plot aids in identifying the model with the best training and validation balance.

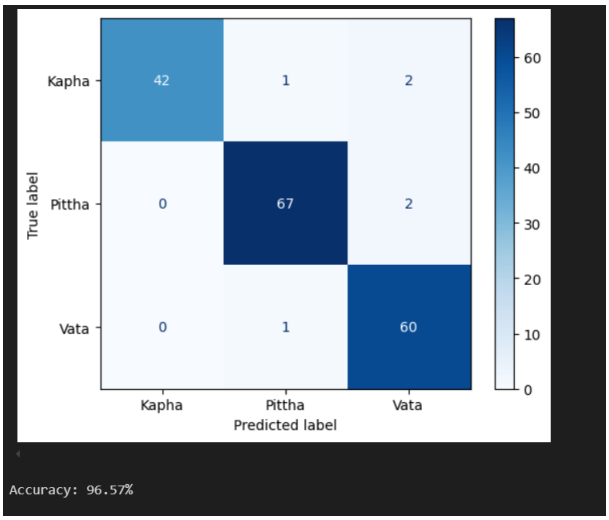


Figure 4.7: Model Evaluation

Figure 4.7 illustrates the confusion matrix for evaluating the model performance on the Skin Dataset, where the skin types (dry, normal, oily) are mapped to the doshas (Vata, Pitta, Kapha). The confusion matrix compares the true labels with the predicted labels, highlighting the true positives, false positives, false negatives, and true negatives, offering a clear view of the model's classification accuracy. The same approach will be followed for evaluating the other datasets Eyes, Lips, Face Shape, and Hair to assess model performance in mapping image features to doshas across all categories.

Confusion Matrix Analysis for Skin Dataset

The confusion matrix (Figure 4.7) above illustrates the evaluation of the model performance on the Skin Dataset. The dataset comprises three skin types (Dry, Normal, Oily), which are mapped to the three doshas (Vata, Pitta, Kapha). The matrix compares the true labels with the predicted labels, offering insight into the model's classification accuracy by highlighting true positives, false positives, false negatives, and true negatives.

Predicted \Actual	Kapha	Pitta	Vata
Kapha	42	1	2
Pitta	0	67	2
Vata	0	1	60

Table 4.7: Confusion Matrix for Skin Dataset

Detailed Analysis

- **Total Test Dataset Size:**
  The total number of samples in the test dataset is the sum of all entries in the confusion matrix:

$$42+1+2+0+67+2+0+1+60=175 \text{ images.}$$

- **Correctly Classified Samples:**
  The diagonal elements represent the correctly classified samples:

$$42 \text{ (Kapha)}, \ 67 \text{ (Pitta)}, \text{ and } 60 \text{ (Vata)}.$$

  Total correctly classified samples:

$$42+67+60=169.$$

- **Misclassified Samples:**
  The off-diagonal elements indicate misclassified samples:
  - Kapha misclassified as Pitta: 1.
  - Kapha misclassified as Vata: 2.
  - Pitta misclassified as Vata: 2.
  - Vata misclassified as Pitta: 1.

Total misclassified samples:

$$1+2+2+1=6.$$

- **Classification Accuracy:**

  The classification accuracy is calculated as:

$$\text{Accuracy} = \frac{\text{Correctly Classified Samples}}{\text{Total Samples}} = \frac{169}{175} \approx 96.57\%.$$

- **Per-Class Accuracy:**

  - **Kapha:** $\frac{42}{42+1+2} = \frac{42}{45} \approx 93.33\%$.
  - **Pitta:** $\frac{67}{67+0+2} = \frac{67}{69} \approx 97.10\%$.
  - **Vata:** $\frac{60}{60+1+0} = \frac{60}{61} \approx 98.36\%$.

4.6 SUMMARY OF THE MODEL TRAINING AND EVALUATION

The model processes 128×128 RGB images, normalizes them, and uses a pre-trained network for initial feature extraction. Convolutional and max-pooling layers detect patterns and reduce spatial dimensions. The extracted features are flattened, passed through a dense layer with 256 neurons, and regularized with dropout to prevent overfitting. The final softmax output layer classifies the image into Kapha, Pitta, or Vata.

This structure is consistently applied across models and datasets, with adjustments made to suit specific requirements while maintaining flexibility and robustness. Leveraging pre-trained networks reduces training time and enhances performance, making it effective for Ayurvedic dosha classification across diverse datasets.

CHAPTER 5

RESULTS AND ANALYSIS

The comparative performance analysis of seven transfer learning models—VGG16, VGG19, ResNet50, MobileNet, DenseNet121, InceptionV3, and EfficientNet—was conducted across five datasets: Hair, Skin, Face Shape, Eyes, and Lips. Performance metrics such as accuracy, precision, recall, and F1-score were visualized using bar charts and graphs. These charts compare the models performance on each dataset. Training and validation accuracy and loss curves were also analyzed to understand the convergence and learning behavior of the models, offering insights into their efficiency across the different datasets.

5.1 PERFORMANCE EVALUATION

The performance of the seven transfer learning models on five distinct datasets will be evaluated using key metrics: Precision, Recall, and F1 Score. These metrics provide insight into the models' accuracy, ability to handle class imbalances, and overall effectiveness in classification tasks.

5.1.1Metrics Used

Precision is the ratio of correctly predicted positive observations to the total predicted positives. It is a measure of the accuracy of the positive predictions made by the model. High precision means that the model correctly identifies most of the positive instances. The equation for precision is given by Equation 1:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{1}$$

Recall is the ratio of correctly predicted positive observations to all the observations in the actual class. It measures the model's ability to identify all relevant instances. The equation for recall is given by Equation 2:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{2}$$

A high recall means that the model successfully identifies most of the positive instances, minimizing false negatives.

The **F1 Score** is the harmonic mean of precision and recall, providing a balance between the two metrics. It is particularly useful when there is an uneven class distribution. The equation for F1 Score is given by Equation 3:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

The F1 Score is a measure of the model's overall effectiveness in handling both precision and recall, ensuring that both false positives and false negatives are minimized.

5.1.2 Tabular Representation of the Results

The tables summarize the performance of seven transfer learning models across various datasets, highlighting Precision, Recall, and F1-Score. MobileNet and DenseNet121 typically outperform the other models.

Table 5.1: Metrics Comparison Across Models for Eyes Dataset

Model	Precision	Recall	F1-Score
VGG16	88.01	87.31	87.01
VGG19	86.20	81.03	82.45
RESNET50	19.10	44.23	26.30
MOBILENET	96.20	95.60	96.20
INCEPTIONV3	80.10	76.32	75.21
DENSENET121	94.20	94.23	94.56
EFFICIENT	16.23	39.23	22.24

Table 5.1 shows the Precision, Recall, and F1 Score results for the Eyes dataset across the seven transfer learning models, providing an overview of model performance in classifying the dataset.

Table 5.2: Metrics Comparison Across Models for Lips Dataset

Model	Precision	Recall	F1-Score
VGG16	82.13	81.45	81.25
VGG19	76.14	74.46	73.23
RESNET50	45.56	35.63	32.89
MOBILENET	96.23	96.45	96.75
INCEPTIONV3	76.85	75.45	74.21
DENSENET121	73.25	72.15	72.36
EFFICIENT	11.23	33.21	17.48

Table 5.2 presents the Precision, Recall, and F1 Score results for the Lips dataset across the seven transfer learning models, highlighting the performance of each model in classifying the dataset.

Table 5.3: Metrics Comparison Across Models for Hair Dataset

Model	Precision	Recall	F1-Score
VGG16	93.10	93.10	92.31
VGG19	93.23	93.31	93.21
RESNET50	33.10	42.02	34.71
MOBILENET	95.21	96.20	94.31
INCEPTIONV3	91.23	92.13	91.41
DENSENET121	89.31	88.31	88.21
EFFICIENT	11.75	34.27	17.50

Table 5.3 displays the Precision, Recall, and F1 Score results for the Hair dataset across the seven transfer learning models, providing a comparison of each model's classification performance on this dataset.

Table 5.4: Metrics Comparison Across Models for Skin Dataset

Model	Precision	Recall	F1-Score
VGG16	83.82	82.35	81.95
VGG19	77.25	70.59	71.26
RESNET50	08.65	29.41	13.37
MOBILENET	95.20	97.20	96.50
INCEPTIONV3	82.70	76.47	72.41
DENSENET121	88.24	88.24	88.24
EFFICIENT	12.46	35.29	18.41

Table 5.4 presents the Precision, Recall, and F1 Score results for the Skin dataset across the seven transfer learning models, offering a comparison of the models' effectiveness in classifying skin types.

Table 5.5: Metrics Comparison Across Models for Face Dataset

Model	Precision	Recall	F1-Score
VGG16	95.21	94.87	94.92
VGG19	90.97	88.89	89.13
RESNET50	81.62	73.50	74.97
MOBILENET	99.17	99.15	99.15
INCEPTIONV3	95.55	94.87	94.95
DENSENET121	43.31	65.81	52.24
EFFICIENT	96.26	95.73	95.80

Table 5.5 displays the Precision, Recall, and F1 Score results for the Face Shape dataset across the seven transfer learning models, providing a comparative analysis of the models' classification performance.

5.1.3Training and Validation Metrics

Training and validation metrics help assess model performance, with a gap indicating overfitting. Monitoring them ensures better generalization.

5.2 ACCURACY AND LOSS TRENDS OF EYES DATASET

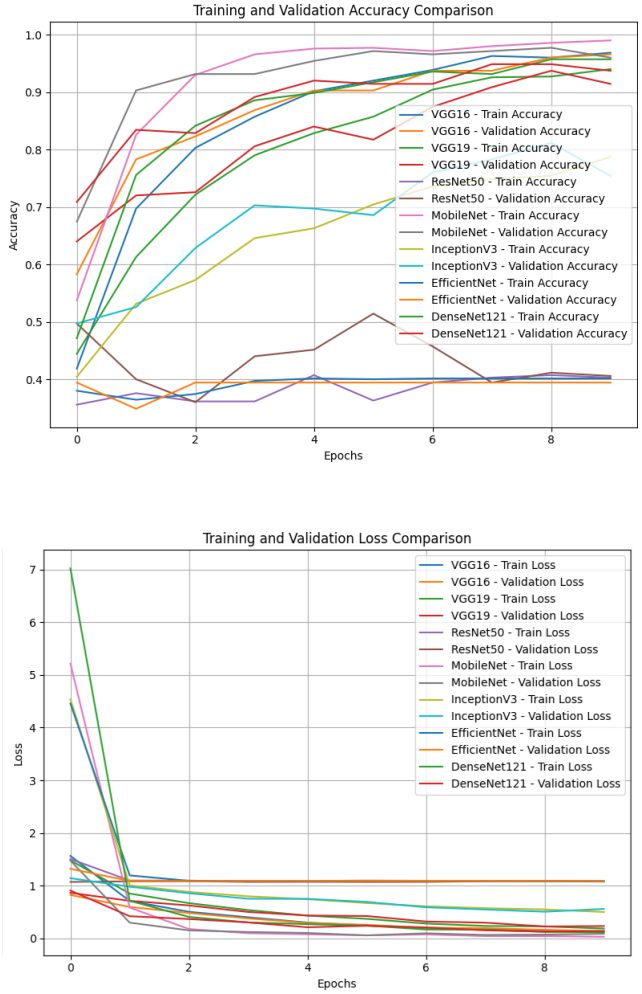


Figure 5.1: Graph Comparison of Eyes Dataset

Figures 5.1 and 5.2 highlight MobileNet as the top performer, with the highest validation accuracy (0.96) and the lowest validation loss (0.09). VGG16, VGG19, and ResNet50 show moderate validation accuracy (0.80 to 0.97) and higher validation loss, indicating potential overfitting. InceptionV3, EfficientNet, and DenseNet121 have lower accuracy and higher loss, suggesting underperformance.

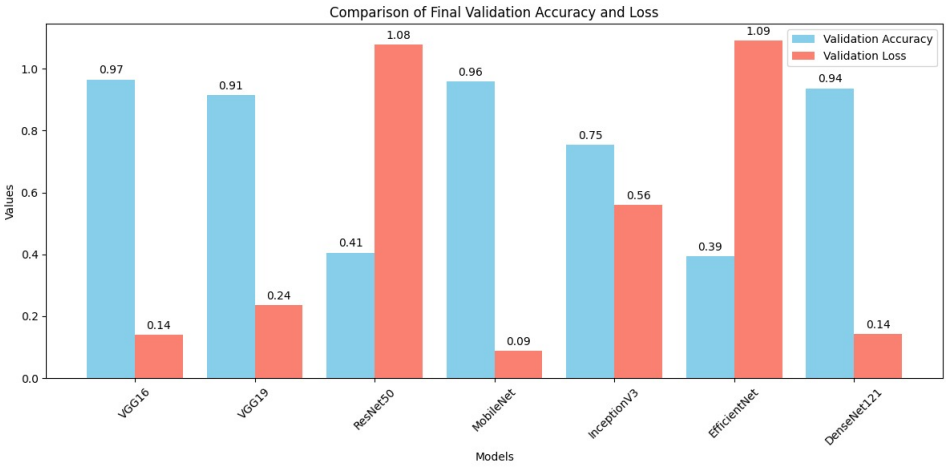
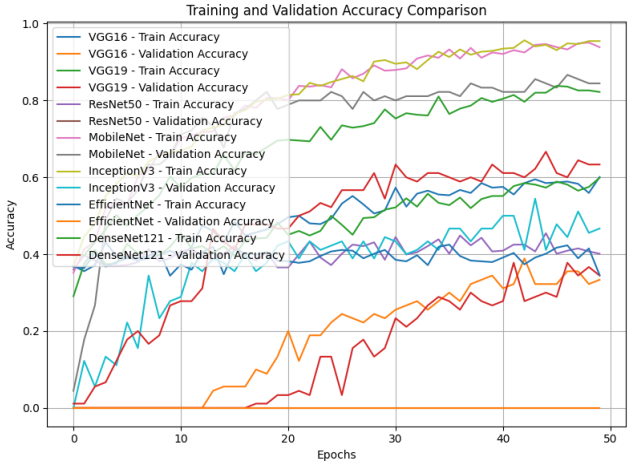


Figure 5.2: Performance Analysis on Eye Data

5.3 ACCURACY AND LOSS TRENDS OF LIPS DATASET



Figures 5.3 and 5.4 present the final validation accuracy and loss values, reinforcing the trends from earlier graphs. MobileNet outperforms all models, with the highest validation accuracy at 0.84 and the lowest validation loss at 0.57. VGG16, VGG19, and ResNet50 show moderate accuracy (0.33 to 0.34) and higher loss (1.20 to 1.41), suggesting overfitting. InceptionV3, EfficientNet, and DenseNet121 have lower accuracy (below 0.50) and higher loss (above 0.97), indicating potential issues with convergence or optimization.

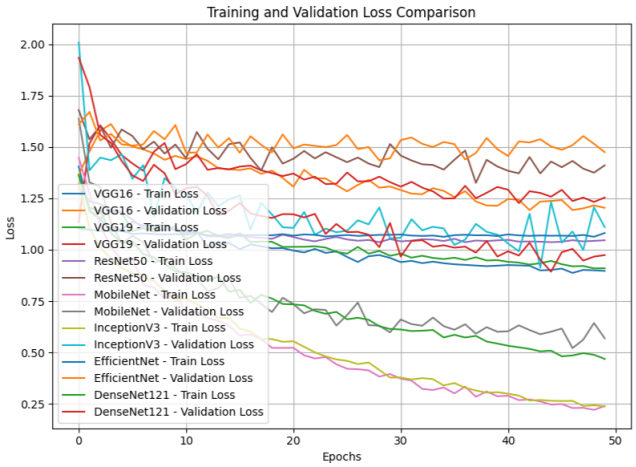


Figure 5.3: Graph Comparison of Lips Dataset

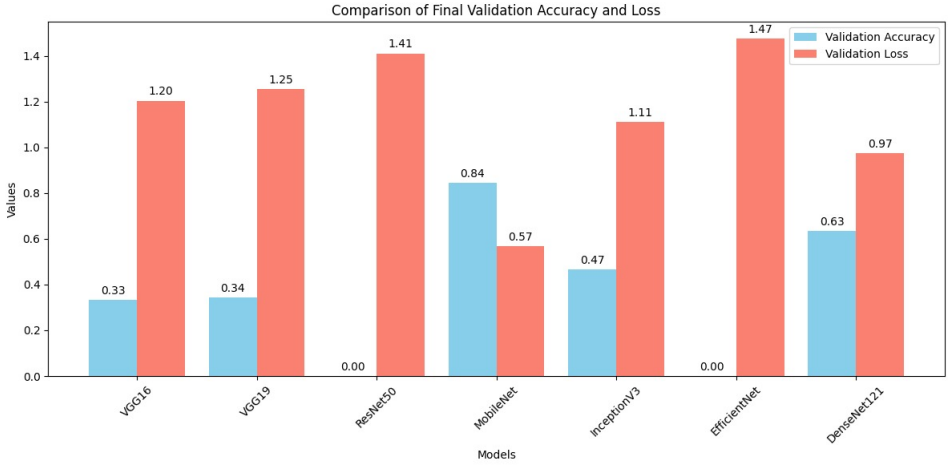


Figure 5.4: Performance Analysis on Lips Data

5.4 ACCURACY AND LOSS TRENDS OF FACE DATASET

Figures 5.5 and 5.6 demonstrate that MobileNet significantly outperforms all other models in terms of validation accuracy (0.95), precision, recall, and F1 score. This indicates its superior ability to generalize across various datasets and effectively classify different skin types. On the other hand, VGG16, VGG19, and ResNet50 show moderate performance, with validation accuracy ranging from 0.80 to 0.83, and higher validation loss, which may suggest potential overfitting and inadequate generalization. InceptionV3,

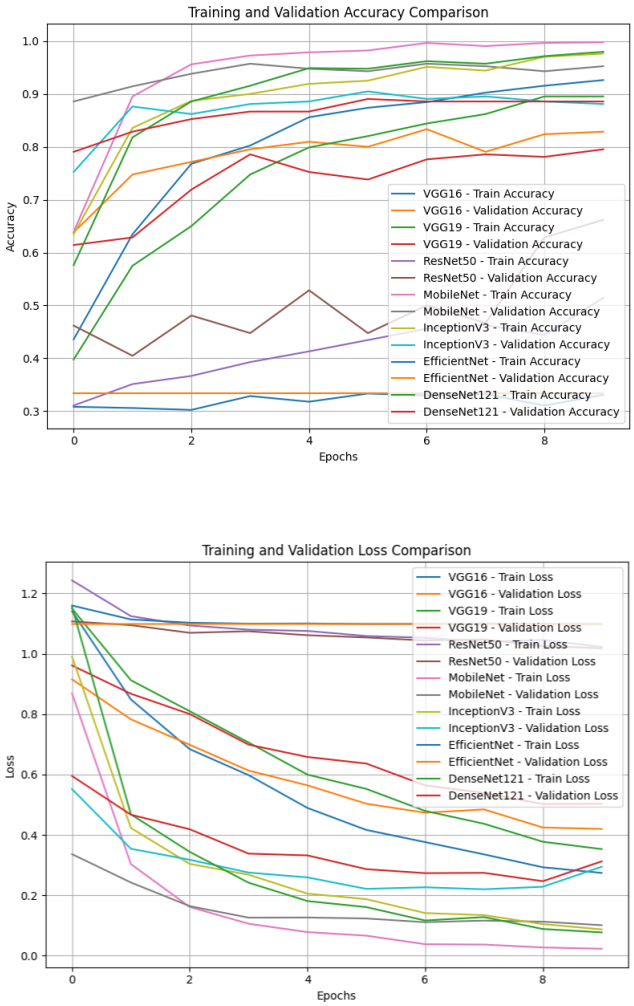


Figure 5.5: Graph Comparison of Face Dataset

EfficientNet, and DenseNet121 display even lower accuracy (below 0.33) and higher validation loss, pointing to challenges with convergence, optimization, or the need for more extensive fine-tuning to improve their performance. These results highlight MobileNet's efficiency and robustness in comparison to the other models in handling skin classification tasks.

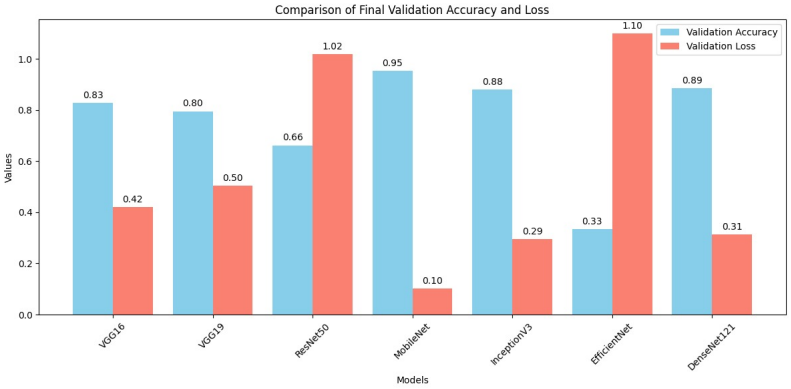
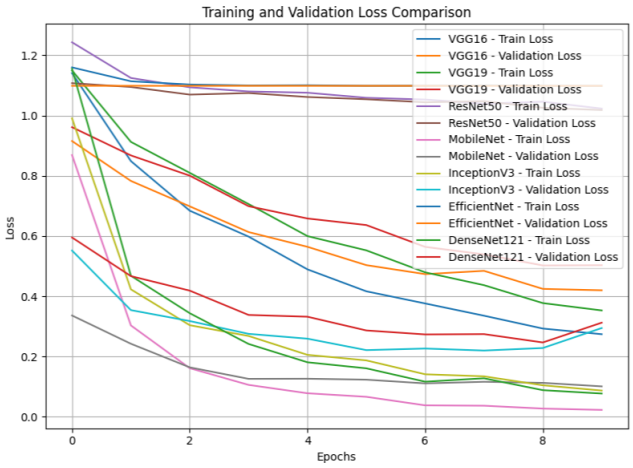


Figure 5.6: Performance Analysis on Face Data

5.5 ACCURACY AND LOSS TRENDS OF HAIR DATASET



Figures 5.7 and 5.8 highlight MobileNet's strong performance, with a validation accuracy of 95.44 , confirming it as the best model. Compared to VGG16, VGG19, and ResNet50, which show moderate accuracy and higher loss, MobileNet demonstrates better generalization. InceptionV3, EfficientNet, and DenseNet121 also show lower performance, further supporting MobileNet's superior balance between accuracy and efficiency.

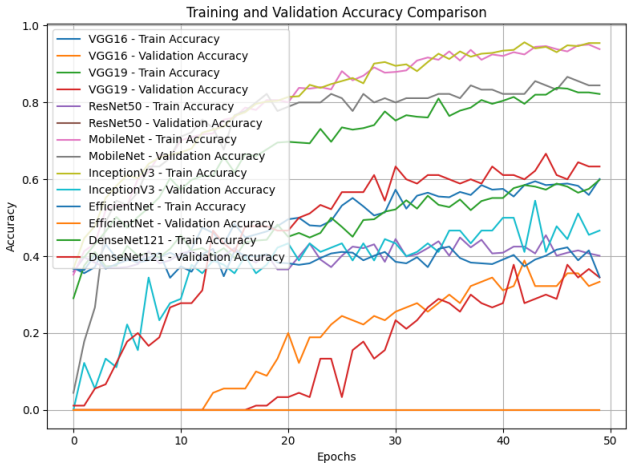


Figure 5.7: Graph Comparison of Hair Dataset

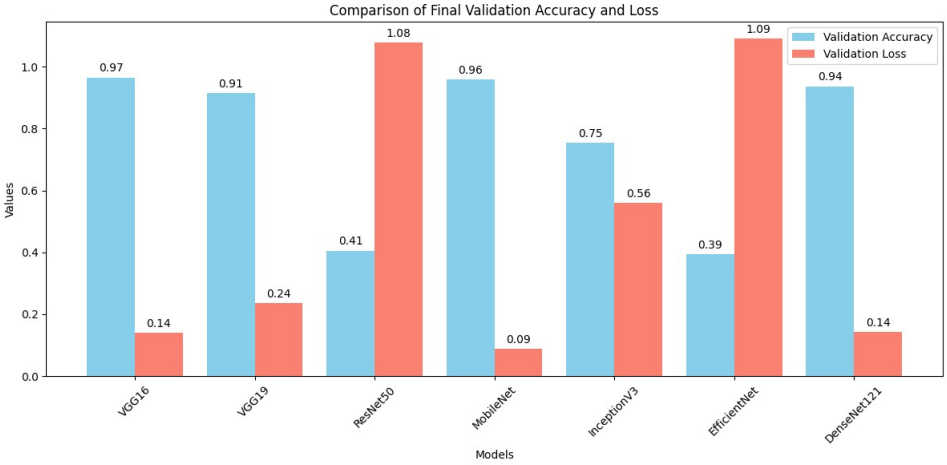


Figure 5.8: Performance Analysis on Hair Data

5.6 ACCURACY AND LOSS TRENDS OF SKIN DATASET

The skin dataset includes images classified into different skin types, such as dry, normal, and oily, to aid in the development of models for skin classification and analysis. This dataset is used to assess the performance of various machine learning algorithms in predicting skin conditions based on visual features.

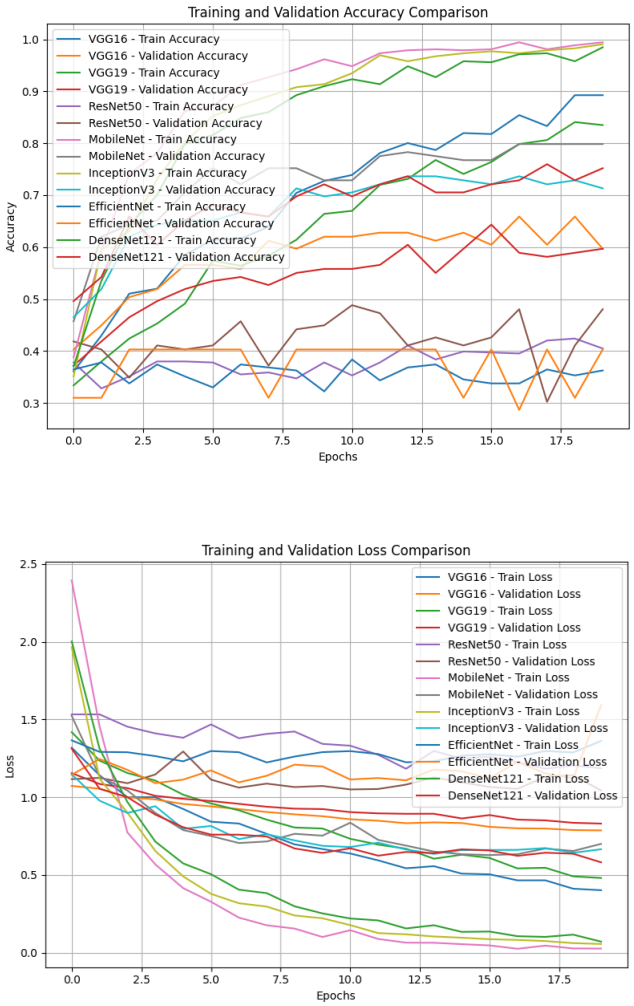


Figure 5.9: Graph Comparison of Skin Dataset

Figures 5.9 and 5.10 present the final evaluation metrics, including precision, recall, and F1 score, for each model. MobileNet outperforms the others with the highest validation accuracy (0.80) and lowest validation loss (0.40). VGG16, VGG19, and ResNet50 show moderate accuracy (0.60) and higher loss (0.70), indicating overfitting. InceptionV3, EfficientNet, and DenseNet121 exhibit lower accuracy (below 0.75) and higher loss (above 0.50), suggesting a need for optimization.

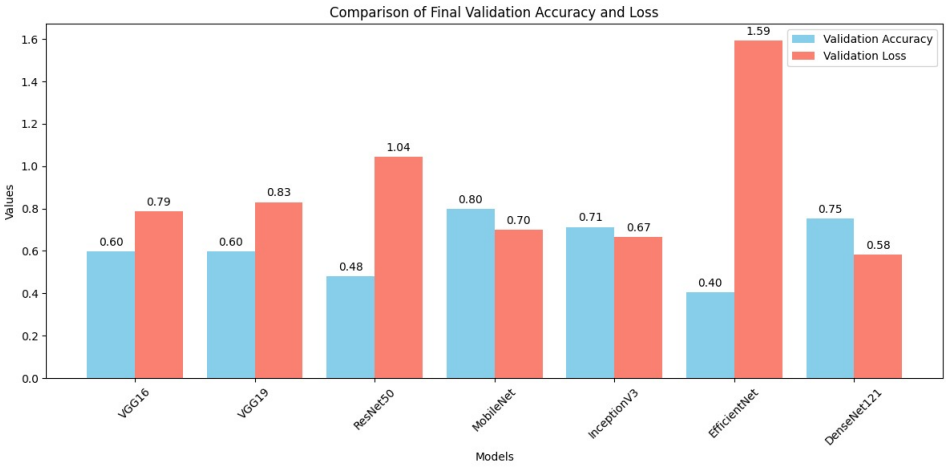


Figure 5.10: Performance Analysis on Skin Data

5.7 INFERENCES FROM TRANSFER LEARNING MODEL EVALUATION

In conclusion, the transfer learning model evaluation reveals that MobileNet consistently outperforms other architectures, demonstrating the highest validation accuracy (95.44) and the lowest validation loss. Its strong generalization capabilities make it the most efficient in terms of performance and computational efficiency. While VGG16, VGG19, and ResNet50 show moderate performance with signs of potential overfitting, InceptionV3, EfficientNet, and DenseNet121 exhibit lower performance and higher validation loss, suggesting a need for further fine-tuning or optimization. Overall, MobileNet emerges as the top transfer learning model for this task, offering the best balance between accuracy and efficiency. Additionally, MobileNet's lightweight design makes it ideal for real-time applications, and its robustness highlights the importance of training with diverse datasets to improve generalization. Future work could include exploring ensembling techniques to further optimize performance.

CHAPTER 6

CONCLUSION AND FUTURE WORK

The study evaluated the performance of various transfer learning models in Ayurveda dosha classification using small datasets. The focus was on understanding how models like VGG16, VGG19, ResNet-50, MobileNet, InceptionV3, and DenseNet121 perform when classifying Ayurveda doshas based on skin images

6.1 MODEL PERFORMANCE SUMMARY ACROSS DATASETS

Various transfer learning models, including MobileNet, VGG16, VGG19, ResNet50, InceptionV3, EfficientNet, and DenseNet121, were evaluated for classifying Dosha types in Ayurveda: Vata, Pitta, and Kapha. These models were trained and tested across diverse datasets, including hair, skin, face shape, eyes, and lips, with performance measured using validation accuracy, validation loss, precision, recall, and F1 score. MobileNet consistently outperformed the other models, achieving the highest validation accuracy of 95.44 and the lowest validation loss, making it the most efficient model in terms of both performance and computational cost. VGG16, VGG19, and ResNet50 demonstrated moderate accuracy, but their tendency toward overfitting highlighted the need for further refinement. InceptionV3, EfficientNet, and DenseNet121 showed lower performance, suggesting that these models may require more optimization.

MobileNet's strong generalization capabilities make it the best choice for the task, effectively balancing accuracy and efficiency. This study contributes to the integration of deep learning with traditional Ayurvedic

practices, paving the way for personalized health assessments. The results offer valuable insights into how transfer learning models can enhance the application of Ayurveda in modern healthcare. Future efforts will focus on developing a robust real-time diagnostic tool for Dosha identification, ultimately advancing personalized healthcare solutions based on Ayurvedic principles.

6.2 FUTURE WORK

The future work of this project lies in the integration of Prakriti (natural constitution) and Vikriti (current imbalances) analysis to create a unified, AI-driven system for personalized healthcare. By combining these analyses, the project will enable a holistic understanding of an individual's health, addressing both their inherent tendencies and current health conditions. The system will feature an AI-powered chatbot as the primary interface, allowing users to report symptoms, receive personalized recommendations, and access dynamic health insights tailored to their Prakriti and Vikriti profiles. Real-time monitoring through wearable devices and automated image analysis will continuously assess changes in health, providing proactive care and trend visualization. The chatbot will also offer symptom-based inquiries, reminders, and consultations with Ayurvedic practitioners, ensuring a user-friendly experience. To improve accuracy and scalability, the dataset will be expanded to include diverse populations and health conditions. This project envisions collaboration with Ayurvedic practitioners and telemedicine platforms, promoting preventive healthcare and bridging traditional wisdom with modern technology. Ultimately, the integration of Prakriti and Vikriti analysis will lay the foundation for a personalized, adaptive, and predictive healthcare system, empowering individuals and advancing the global application of Ayurveda.

REFERENCES

[1] Steven Euijong Whang, Yuji Roh, Hwanjun Song, and Jae-Gil Lee. Data collection and quality challenges in deep learning: A data-centric ai perspective. *The VLDB Journal*, 32(4):791–813, 2023.

[2] Hamed Taherdoost. Data collection methods and tools for research; a step-by-step guide to choose data collection technique for academic and business research projects. *International Journal of Academic Research in Management (IJARM)*, 10(1):10–38, 2021.

[3] Yuji Roh, Geon Heo, and Steven Euijong Whang. A survey on data collection for machine learning: a big data-ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1328–1347, 2019.

[4] Dipali Shete, Sachin Bojewar, and Ankit Sanghvi. Survey paper on web content extraction & classification. In *2021 6th International Conference for Convergence in Technology (I2CT)*, pages 1–6. IEEE, 2021.

[5] Ajay Sudhir Bale, Naveen Ghorpade, S Rohith, S Kamalesh, R Rohith, and BS Rohan. Web scraping approaches and their performance on modern websites. In *2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 956–959. IEEE, 2022.

[6] Vidhi Singrodia, Anirban Mitra, and Subrata Paul. A review on web scrapping and its applications. In *2019 international conference on computer communication and informatics (ICCCI)*, pages 1–6. IEEE, 2019.

[7] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[8] K Maharana, S Mondal, and B Nemade. A review: Data pre-processing and data augmentation techniques. global transitions proceedings, 3 (1), 91–99. In *International Conference on Intelligent Engineering Approach (ICIEA-2022)*, 2022.

[9] Teerath Kumar, Rob Brennan, Alessandra Mileo, and Malika Bendechache. Image data augmentation approaches: A comprehensive survey and future directions. *IEEE Access*, 2024.

[10] Puneet Kumar and Dalwinder Singh. Pre-processing techniques for enhancing cnn performance. In *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, pages 1–5. IEEE, 2024.

[11] Agnieszka Mikołajczyk and Michał Grochowski. Data augmentation for improving deep learning in image classification problem. In *2018 international interdisciplinary PhD workshop (IIPhDW)*, pages 117–122. IEEE, 2018.

[12] Gayatri Gadre. Classification of humans into ayurvedic prakruti types using computer vision. 2019.

[13] Vishu Madaan and Anjali Goyal. Predicting ayurveda-based constituent balancing in human body using machine learning methods. *IEEE Access*, 8:65060–65070, 2020.

[14] Sanyam Jain and Taruna Chawla. Ayurvedic doshas identification using face and body image features-a review. *International Journal of Advanced Research in Computer Science*, 12(3):19–22, 2021.

[15] Lakshmi Bheemavarapu and K Usha Rani. A review on role of data science in ayurveda based disease diagnosis using prakriti type in trividha pariksha. *Information Technology in Industry*, 9(3):1038–1048, 2021.

[16] Pradeep Tiwari, Rintu Kutum, Tavpritesh Sethi, Ankita Shrivastava, Bhushan Girase, Shilpi Aggarwal, Rutuja Patil, Dhiraj Agarwal, Pramod Gautam, Anurag Agrawal, et al. Recapitulation of ayurveda constitution types by machine learning of phenotypic traits. *PloS one*, 12(10):e0185380, 2017.

[17] Harpreet Singh, Sapna Bhargava, Sailesh Ganeshan, Ravneet Kaur, Tavpritesh Sethi, Mukesh Sharma, Madhusudan Chauhan, Neerja Chauhan, Rishipal Chauhan, Partap Chauhan, et al. Big data analysis of traditional knowledge-based ayurveda medicine. *Progress in Preventive Medicine*, 3(5):e0020, 2018.

[18] Sarra Guefrechi, Marwa Ben Jabra, Adel Ammar, Anis Koubaa, and Habib Hamam. Deep learning based detection of covid-19 from chest x-ray images. *Multimedia tools and applications*, 80:31803–31820, 2021.

[19] Sheldon Mascarenhas and Mukul Agarwal. A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification. In *2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON)*, volume 1, pages 96–99. IEEE, 2021.

[20] Kayiram Kavitha, Prashant Sharma, Shubham Gupta, and RVS Lalitha. Medicinal plant species detection using deep learning. In *2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT)*, pages 01–06. IEEE, 2022.

[21] Rachna Jain, Preeti Nagrath, Gaurav Kataria, V Sirish Kaushik, and D Jude Hemanth. Pneumonia detection in chest x-ray images using convolutional neural networks and transfer learning. *Measurement*, 165:108046, 2020.

[22] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[23] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[25] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: efficient convolutional neural networks for mobile vision applications (2017). *arXiv preprint arXiv:1704.04861*, 126, 2017.

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.