Mini-Project Report On

Facial Acne Classification and Product Recommendation

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science Engineering

By

Navya Rony A (RET20CS143)
Riya Thomas (RET20CS160)
Sandra Sunil (RET20CS179)
Tessa Pratheesh (RET20CS202)

Under the guidance of
Ms. Jincy J. Fernandez

Department of Computer Science & Engineering
Rajagiri School of Engineering and Technology (Autonomous)
Rajagiri Valley, Kakkanad, Kochi, 682039

July 2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RAJAGIRI SCHOOL OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
RAJAGIRI VALLEY, KAKKANAD, KOCHI, 682039

CERTIFICATE

This is to certify that the mini-project report entitled ***"Facial Acne Classification and Product Recommendation"*** *is a bonafide work done by* ***Navya Rony A (RET20CS143), Riya Thomas (RET20CS160), Sandra Sunil (RET20CS179), Tessa Pratheesh (RET20CS202)*** *submitted to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2022-2023.*

Dr. Preetha K. G.
Head of Department
Dept. of CSE
RSET

Ms. Anita John
Mini-Project Coordinator
Asst. Professor
Dept. of CSE
RSET

Ms. Jincy J. Fernandez
Mini-Project Guide
Asst.Professor
Dept. of CSE
RSET

ACKNOWLEDGEMENTS

We are extremely honored to thank Prof. (Dr) P.S. Sreejith, Principal, RSET, who has always made sure that our ladder to success was leaning against the right wall. We thank him for his constant help and support. We thank Dr. Preetha K.G., HoD, Department of Computer Science & Engineering, RSET, whose help and guidance have been a major factor in completing our journey.

We express our gratitude to our Project Coordinator and Class-In-Charge, Ms. Anita John, Assistant Professor, Department of Computer Science and Engineering, and Mr. Sajanraj T D, Assistant Professor, Department of Computer Science and Engineering, for their valuable support. We extend our sincere and heartfelt thanks to our guide Ms. Jincy J. Fernandez, Assistant Professor, Department of Computer Science & Engineering, RSET, for taking the time and effort to review our work and providing valuable advice and feedback from time to time. Last but not least, we would like to express our heartfelt and sincere gratitude to our family and friends for their constant support throughout this entire journey.

Navya Rony A,
Riya Thomas,
Sandra Sunil,
Tessa Pratheesh

ABSTRACT

Acne is a common skin condition that affects a significant portion of the population. Accurate classification of acne severity and personalized product recommendation can greatly assist in its management. In this report, we explore the application of machine learning techniques to classify acne severity and recommend suitable products based on individual skin types. We leverage a dataset of labeled images representing different acne severities and product attributes to train and evaluate our models.

For acne severity classification, we employ convolutional neural networks (CNNs) to extract meaningful features from images. We preprocess the data by resizing and normalizing the images. We train the CNN models using the labeled data and evaluate their performance using standard evaluation metrics such as accuracy, precision, recall, and F1 score. The results demonstrate the effectiveness of the proposed approach in accurately classifying acne severity levels.

To provide personalized product recommendations, we analyze product attributes and customer preferences. We employ collaborative filtering and content-based filtering techniques to create a recommendation system. The system takes into account user profiles, skin type information, and product characteristics to generate tailored recommendations.

Our findings reveal that machine learning models can effectively classify acne severity with high accuracy. The personalized product recommendation system demonstrates its capability to suggest suitable acne products based on individual skin types and preferences. The report concludes with insights into the strengths and limitations of the proposed approach and discusses potential avenues for future research.

Contents

List of Figures

Chapter 1

INTRODUCTION

1.1 General Background

1.1.1 Facial Acne Classification and Product Recommendation

Acne is a common skin condition that affects millions of individuals worldwide. It is characterized by the presence of various types of lesions, such as blackheads, whiteheads, papules, pustules, and cysts. Acne can have a significant impact on a person's physical appearance and self-esteem, particularly when it occurs on the face.

Acne classification involves the categorization of acne severity into different levels, such as mild, moderate, and severe. This classification is crucial in determining appropriate treatment strategies and monitoring progress over time. Traditional methods of acne severity assessment rely on visual examination by dermatologists, which can be subjective and time-consuming. By leveraging machine learning algorithms, it is possible to automate this process and provide accurate and consistent acne severity classification.

Product recommendation for acne management involves suggesting suitable skincare products based on an individual's skin type, concerns, and preferences. With the wide range of available products in the market, personalized recommendations can greatly assist individuals in selecting the most effective and compatible solutions for their specific acne condition. Machine learning techniques enable the analysis of various factors, including product attributes, customer profiles, and feedback, to generate tailored product recommendation.

1.1.2 Convolutional Neural Networks and EfficientNet B0

Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed for image processing and analysis. CNNs have revolutionized computer vision tasks and have shown great effectiveness in various applications, including facial recognition, object detection, and medical image analysis.

CNNs leverage the concept of convolution, which involves applying filters (also called kernels) to input images. Convolutional layers in a CNN learn to detect and extract different features from the input images. As the network deepens, the learned features become increasingly complex and abstract, capturing higher-level representations.

Training a CNN involves feeding labeled training data through the network and updating the network's parameters (weights and biases) using backpropagation and gradient descent algorithms. The network learns to automatically extract and refine features from the input images, gradually improving its ability to classify or analyze visual data. The figure below (figure 1.1) shows the CNN architecture.

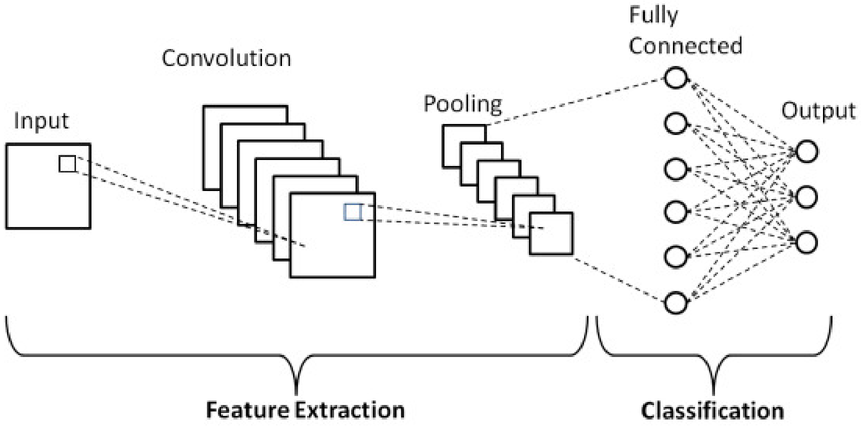


Figure 1.1: CNN Architecture

EfficientNet is a family of convolutional neural network (CNN) architectures that have achieved state-of-the-art performance while being computationally efficient. The EfficientNet model family is based on the concept of scaling neural networks in a principled manner to achieve better performance.

EfficientNet B0 consists of convolutional layers, skip connections, and pooling layers to capture features at multiple scales. It also utilizes efficient depthwise separable convolutions, which reduce the number of parameters and computational cost. This model has demonstrated impressive performance across various computer vision tasks, such as image classification, object detection, and semantic segmentation. By using EfficientNet B0 on facial acne classification and product recommendation, we can benefit from its strong performance in image analysis tasks while keeping the computational demands relatively low. EfficientNet B0 provides a good balance between accuracy and computational efficiency, making it suitable for a wide range of applications where resource constraints may be a consideration. The figure below (figure 1.2) shows the basic architecture of EfficientNet B0.

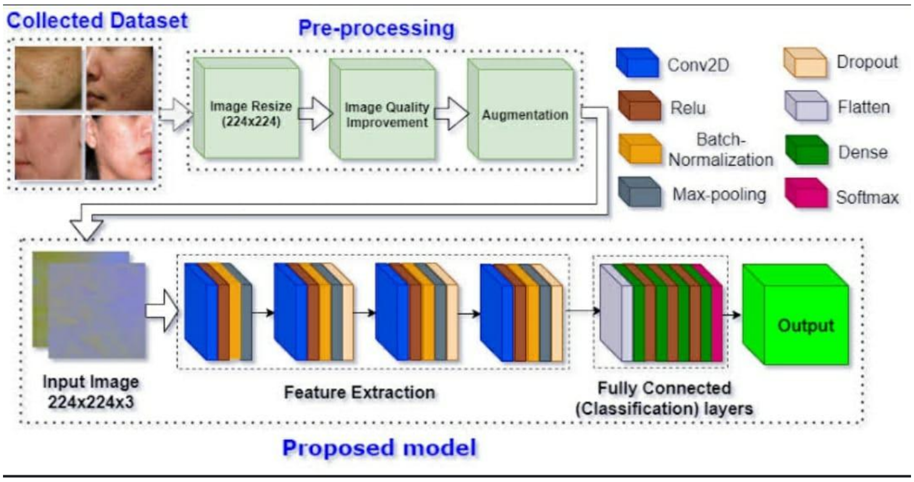


Figure 1.2: Basic Architecture

1.2 Objective

Facial Acne Classification: The first objective of our project is to develop an accurate and automated system for facial acne classification. This involves training a machine learning model, such as a Convolutional Neural Network (CNN), to analyze facial images and classify different levels of acne lesions. The system should be able to detect and categorize acne into low, moderate or severe accurately and objectively. The goal is to provide a reliable and efficient tool for dermatologists and skincare professionals to assess acne severity and plan appropriate treatments.

Product Recommendation: The second objective of our project is to integrate acne classification with a product recommendation system. Based on the individual's specific acne characteristics and skin type the system should recommend personalized skincare products that are suitable for managing acne. The product recommendation component takes into account factors such as ingredients and user skin type to suggest appropriate skincare products that can complement acne treatment regimens.

1.3 Motivation

Improving Acne Diagnosis: Acne is a prevalent skin condition that affects a significant portion of the population. However, the process of diagnosing and assessing acne severity can be subjective and prone to inter-observer variability. By developing an automated acne classification system, we aim to enhance the accuracy and objectivity of acne diagnosis, providing dermatologists and skin care professionals with a reliable tool for assessing acne lesions and determining appropriate treatments.

Enhancing Skincare Personalization: Skincare is not a one-size-fits-all solution. Individuals have different skin types, concerns, and treatment goals. By integrating facial acne classification with a product recommendation system, you seek to personalize skincare recommendations based on the individual's unique acne characteristics. This motivation stems from the understanding that personalized skincare regimens have the potential to yield better treatment outcomes and improve the overall skin care experience for individuals dealing with acne.

Automation for Efficiency and Scalability: Manual assessment and classification of acne can be time consuming and labor-intensive. By automating the acne classification process using machine learning techniques, we aim to increase efficiency, enabling faster and more consistent assessments. Furthermore, an automated system can easily scale to handle large volumes of facial images, making it suitable for research, clinical settings, or even consumer-facing applications.

Potential for Improved Treatment Planning: Accurate classification of acne lesions and understanding their severity is crucial for developing effective treatment plans. By providing dermatologists and skincare professionals with a reliable and objective acne classification system, we aim to contribute to improved treatment planning. This, in turn, can lead to more targeted and tailored interventions, resulting in better patient outcomes.

Empowering Individuals with Information: Skincare is becoming increasingly personalized, with individuals seeking knowledge and control over their skincare routines. By recommending suitable skincare products based on acne characteristics, we aim to empower individuals to make informed decisions about their skincare. This motivation aligns with the growing demand for personalized skincare solutions and the desire to promote self-care and confidence among individuals dealing with acne-related concerns.

1.4 Summary of Report

The primary goal of the project is to develop an efficient model for the accurate classification of acne severity. Throughout this project, we impose the training of a Convolution Neural Network (CNN) to classify the input image as one among the 3 classes, by training the CNN.Thus the system should be able to produce accurate classification among the 3 classes - low, moderate or severe.

Chapter 2

LITERATURE SURVEY

2.1 Deep Learning Object Detection Models

Amandip Sangha1; Mohammad Rizvi1; Detection of acne by deep learning object detection, 2021

This research paper is focused on using deep learning object detection models to predict and detect acne lesions in photographs of patients' faces.

Acne Overview: Acne is a common chronic inflammatory skin condition that affects a significant portion of the global population.

Detection Model: The research employs a deep learning object detection model, specifically YOLOv5, which is a convolutional neural network. The YOLOv5 model has been pre-trained on a dataset called COCO9. Object detection is the process of identifying specific objects within images and predicting rectangular bounding boxes around them.

Dataset: The researchers use a dataset called ACNE043, which consists of 1457 photographs of patients' faces with acne lesions manually annotated by experts. The dataset is split into training and testing subsets, with a focus on training the model to accurately detect and classify acne lesions.

Evaluation Metrics: The performance of the model is evaluated using various metrics, including Intersection over Union (IoU), precision, recall, and the Precision-Recall curve. These metrics measure the accuracy of bounding box predictions and the ability to correctly detect acne lesions.

Model Performance Evaluation: The researchers manually assess the model's predictions on the test set using criteria such as the number of correct detections by the model, the number of incorrect detections, and the number of missed detections. Practical accuracy is emphasized over strict bounding box accuracy, as dermatologists require fairly accurate bounding boxes around acne lesions.

2.2 Convolution Neural Network Models

Chai, J.; Zeng, H.; Li, A.; Ngai, E. W.T. Deep learning in computer vision: A critical review of emerging techniques and application scenarios. Mach. Learn. Appl. 2021

This research paper offers insights into the evolution of deep network architectures for computer vision (CV) applications. It introduces a range of influential CNN-based models, each tailored to specific challenges in image processing. Among these models are AlexNet, VGGNet, GoogLeNet, Inception, ResNet, DenseNet, MobileNets, EfficientNet, and RegNet.

AlexNet, a trailblazer, introduced the power of deep learning to CV, achieving remarkable accuracy on ImageNet with a top-1 accuracy of around 37.5%. VGGNet, with its emphasis on depth and small convolutional filters, brought deeper architectures into focus and achieved top-1 accuracy of approximately 74.5%.

GoogLeNet's inception modules improved efficiency, and Inception-v3 further enhanced accuracy to 78.8% on ImageNet. ResNet's revolutionary skip connections led to better gradient flow, resulting in top-1 accuracy of 75.3% for ResNet-152.

DenseNet's dense connectivity boosted feature reuse, and DenseNet-201 reached top-1 accuracy of 77.3%. MobileNets prioritized efficiency, with MobileNetV2 offering a good trade-off between accuracy and latency.

EfficientNet, which optimizes depth, width, and resolution, set new standards with Efficient Net-B7 achieving top-1 accuracy of 84.4% on ImageNet. RegNet's simplicity and speed yielded competitive performance, often outperforming other models with greater efficiency.

Each model's accuracy and unique design contribute to the CV landscape, offering solutions for diverse challenges and pushing the boundaries of accuracy and efficiency in image classification tasks.

2.3 Acne Grading and Acne Detection

Hang Zhang ; and Tianyi Ma ; Acne Detection by Ensemble Neural Networks , 2022

Acne Grading

Acne grading is a specialized form of image classification focused on estimating the severity of facial acne through facial images. Previous approaches involve utilizing neural networks to classify acne severity, where the severity serves as the label. Shen et al. introduced binary and septenary classifiers to automate acne diagnosis, categorizing acne into seven classes. The binary classifier distinguished skin patches using features from a pre-trained VGG16 model. In a similar vein, Zhao et al. designed a lightweight model for assessing acne severity in mobile phone selfies, employing image rolling augmentation and OpenCV-based segmentation. Yang et al. also partitioned clinical images into regions to evaluate acne severity. Lim et al. created an automated system for calculating the Investigator's Global Assessment (IGA) scale by adapting three convolutional neural networks (DenseNet, Inception v4, and ResNet18) to the task. Wu et al. addressed data scarcity by introducing the ACNE04 dataset, annotated with acne severity and lesion bounding boxes, using ResNet-50 with a Gaussian-based regression approach. Liu et al. improved severity prediction with an ensemble classification framework, AcneGrader, achieving 85% prediction accuracy and accurate patient diagnoses.

Acne Detection

Acne detection involves locating acne in facial images to aid diagnosis. Rashatapprucksa et al. employed Faster-RCNN and R-FCN to detect acne, reporting up to 28.3% mean average precision (mAP). Sangha et al. fine-tuned YOLOv5 on the ACNE04 dataset, performing well in single-class acne detection but encountering challenges in multi-class detection due to illumination inconsistencies and scale variations. Min et al. introduced ACNet, a network consisting of Composite Feature Refinement, Dynamic Context Enhancement, and Mask-Aware Multi-Attention modules. ACNet outperformed previous methods by Rashataprucksa et al., achieving a 20.5 mAP on the ACNE04 dataset through improved acne detection capabilities.

2.4 EfficientNet B0

Mingxing Tan 1 Quoc V. Le 1; EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, 2020

The research paper discusses a study on model scaling for convolutional neural networks (ConvNets), particularly focusing on a newly developed network architecture called EfficientNet. The key points are as follows:

Importance of Baseline Network: The study emphasizes that having a strong baseline network is crucial, as model scaling doesn't change the core layer operators in ConvNets. They propose a new mobile-size baseline called EfficientNet to demonstrate the effectiveness of their scaling method.

Architecture Search: The authors use a multi-objective neural architecture search technique inspired by previous work. They optimize both accuracy and floating-point operations per second (FLOPS) using specific coefficients to strike a balance between the two.

EfficientNet-B0: The search produces an efficient network architecture called EfficientNet -B0, which is slightly larger than a comparable model but optimized for FLOPS rather than latency.

Scaling Method: The document outlines a two-step scaling method for EfficientNet models. The first step involves finding optimal values for scaling coefficients, and the second step involves using these coefficients to scale up the baseline network to create models EfficientNet-B1 to B7.

Experiments: The study evaluates their scaling method on existing ConvNets and introduces results for EfficientNet models. The results show that EfficientNet models achieve higher accuracy with significantly fewer parameters and FLOPS compared to other ConvNets.

Transfer Learning: EfficientNet models are also evaluated on various transfer learning datasets, demonstrating their strong performance and efficiency compared to other models.

Latency Comparison: Inference latency is measured for several ConvNets, including EfficientNet models, showing that the EfficientNet models are not only accurate but also computationally efficient.

Visualizing Activation Maps: Class Activation Maps (CAMs) are used to visualize how different scaling methods affect the models' focus on relevant object regions. EfficientNet models with compound scaling tend to focus on more detailed regions.

Chapter 3

PROPOSED METHOD

3.1 Problem Definition

To develop a machine learning model that can accurately classify acne based on input images and recommend suitable skincare products for acne treatment. The goal is to provide a reliable and automated solution to assist dermatologists and individuals in identifying and addressing different types of acne, thereby improving the efficiency and effectiveness of acne treatment.

3.2 Methodology

The methodology for acne classification and product recommendation using machine learning in Python typically involves several steps.

3.2.1 Data collection

Gather a comprehensive dataset that includes information on acne severity,skin type, and product effectiveness. The dataset should be diverse and representative of different acne severities.

3.2.2 Data Preprocessing

Clean the dataset by handling missing values, removing irrelevant features, and normalizing or standardizing the data if necessary. This step ensures that the data is in a suitable format for training the machine learning model.

3.2.3 Feature Engineering

Extract or create relevant features from the raw data that can potentially contribute to acne severity classification and product recommendation. This may involve deriving new features from existing ones or applying domain knowledge to enhance the predictive power of the model

3.2.4 Model Selection

Choose an appropriate machine learning algorithm for acne severity classification. Consider the characteristics of the dataset, the interpretability of the model, and the desired performance.

3.2.5 Model Training

Split the dataset into training and validation sets. Use the training set to train the machine learning model on the labeled acne severity data. Adjust the model's hyperparameters to optimize its performance and minimize overfitting.

3.2.6 Model Evaluation

Evaluate the trained model using the validation set to assess its accuracy.This step helps measure the model's performance and ensure it generalizes well to unseen data.

3.2.7 Acne Severity Classification

Use the trained model to predict the acne severity of new, unseen cases. This classification output can help dermatologists in diagnosing and assessing the severity of acne in patients.

3.3 System Architecture

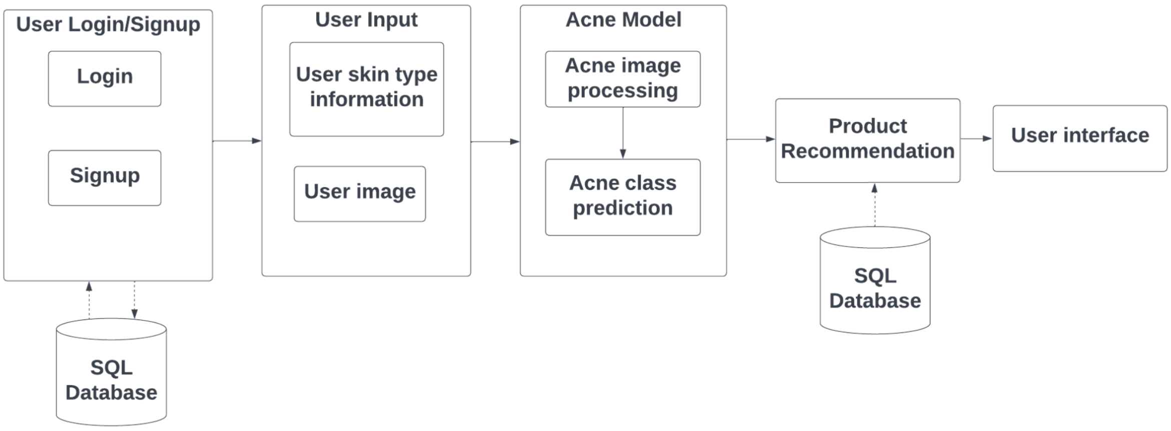


Figure 3.1: System Architecture Diagram

The above figure 3.1 shows the system architecture diagram containing the following modules:

User Login/Signup:

The user can login or signup by entering their credentials. The login information is authenticated against the data store in the SQL database and the signup information is stored in the SQL database.

User Input:

The user uploads the image of their face and their skin type information. The uploaded image is passed to the acne model.

Acne Model:

In the acne model, the user image is preprocessed and the acne class is predicted.

Product Recommendation:

The results from the user input and acne model are used to retrieve the products from the SQL database.

User Interface: The Products retrieved from the database and the acne class predicted by the model are displayed to the user.

3.4 Hardware and Software Requirements

1. SOFTWARE REQUIREMENTS:
    - Operating System: Windows/Linux
    - Simulation System: Google Colab
    - Python 3.x (64-bit version recommended)
    - NumPy library for numerical computing and data manipulation.
    - Scikit-learn library for machine learning algorithms.
    - Pandas library for data analysis and manipulation.
    - Flask library for web application development GUI HTML, CSS, Javascript

2. HARDWARE REQUIREMENTS:
    - A computer with a minimum of 4GB RAM and a multi-core processor.
    - An external storage device to store the skin images and dataset if necessary

3.5 Design of the system

3.5.1 Back-end Design

Internal software data structure

The internal data structures maintained in the program are using the datasets and training the model. The result of the training is validated by passing these results and finding the accuracy.

Global data structure

Each cycle in which the image is passed through the model gives the more accurate predictions.

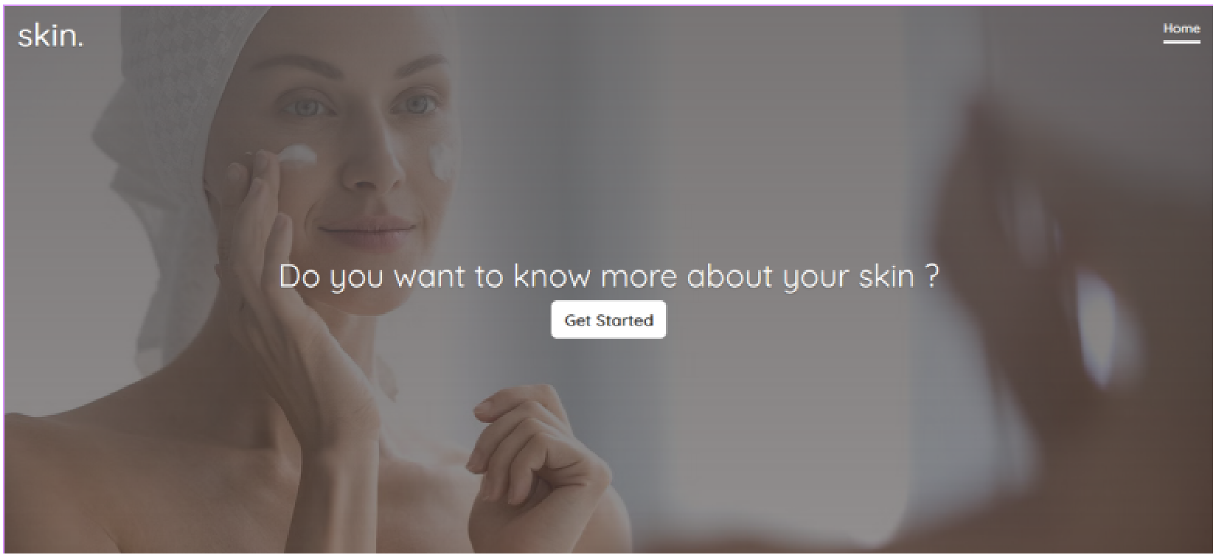
Temporary data structure

Each cycle in which the image is passed through the model gives the more errored predictions.

Database description

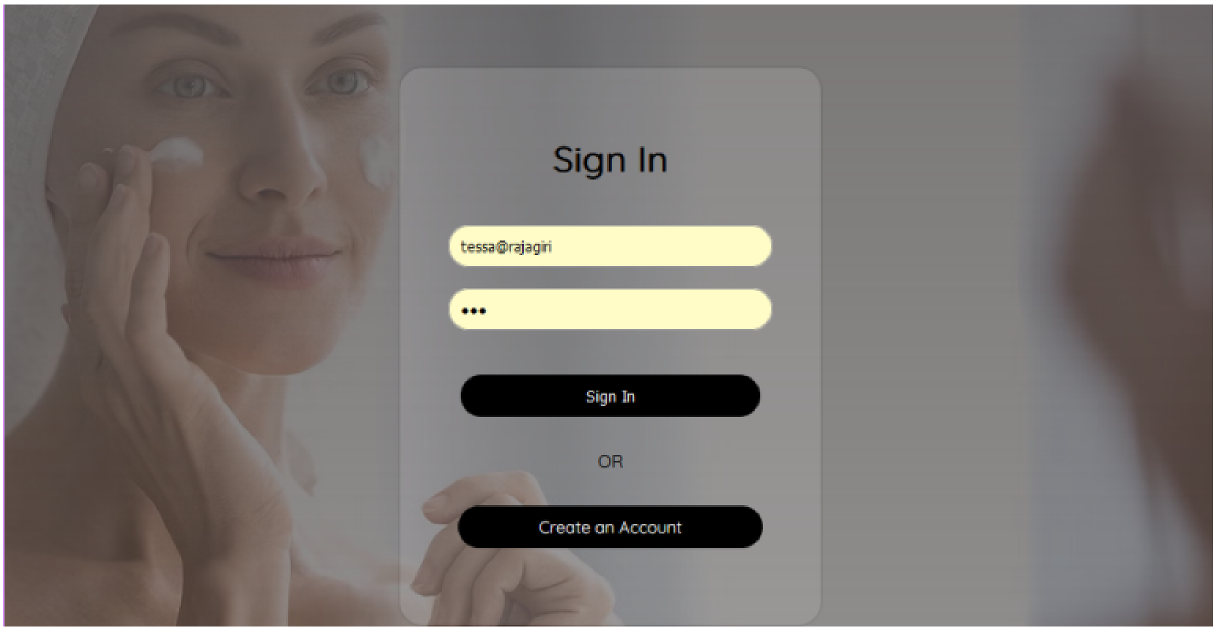
The features of the images are identified and grouped together according to their classes. The weight is initialized to some random value, after each cycle using the Adam the weight is updated and error is reduced, using early stopping when the saturation is reached we stop training.

3.5.2 Front-end Design



Do you want to know more about your skin ?

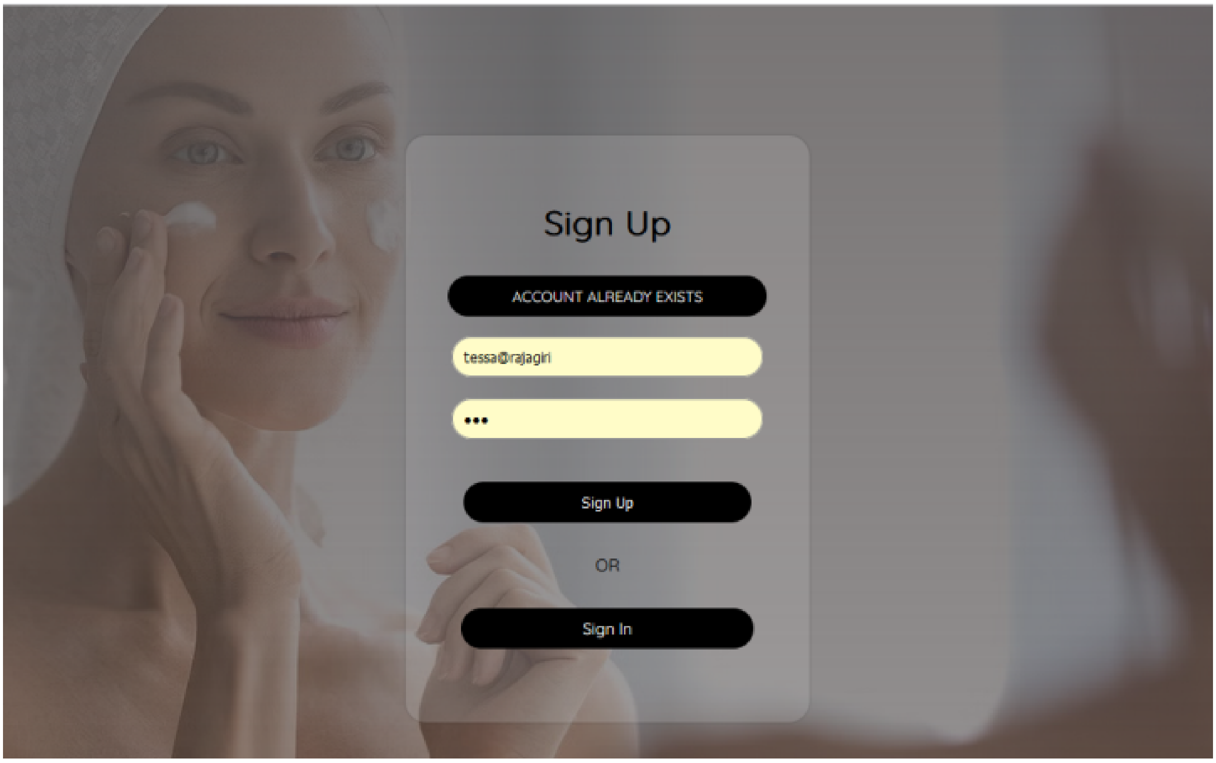
Figure 3.2: Home Page



tessa@rajagiri

•••

Figure 3.3: Sign-in Page



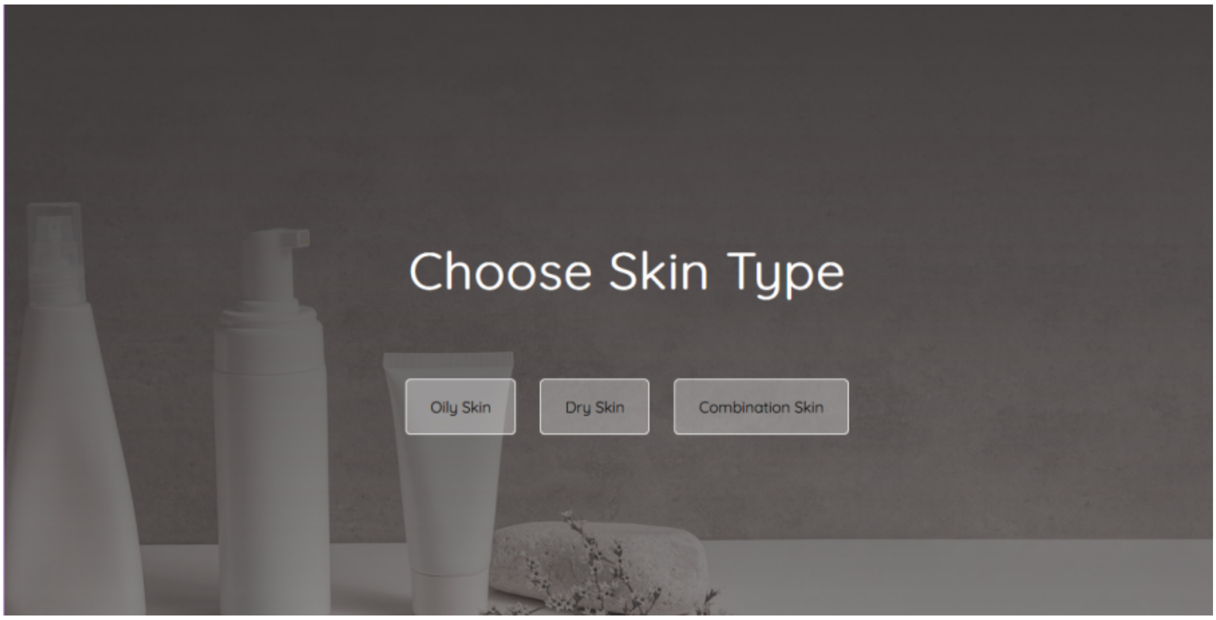
Sign Up

ACCOUNT ALREADY EXISTS

tessa@rajagiri

•••

Figure 3.4: Sign-up Page



Choose Skin Type

Figure 3.5: Skin Type Options Page

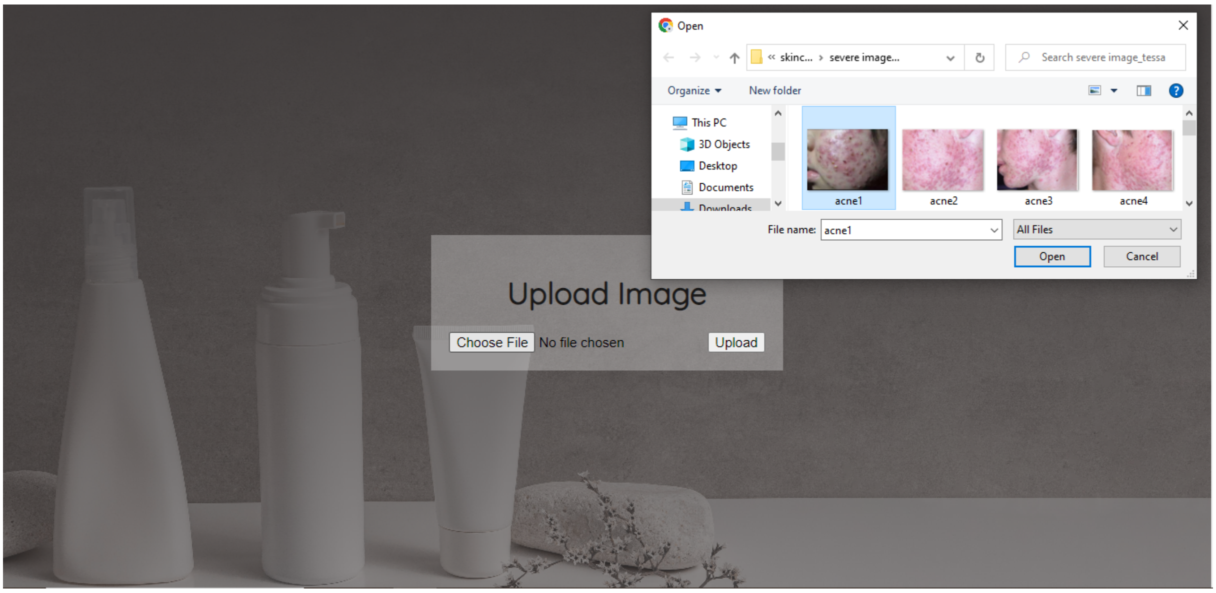


Figure 3.6: Image Selection Page

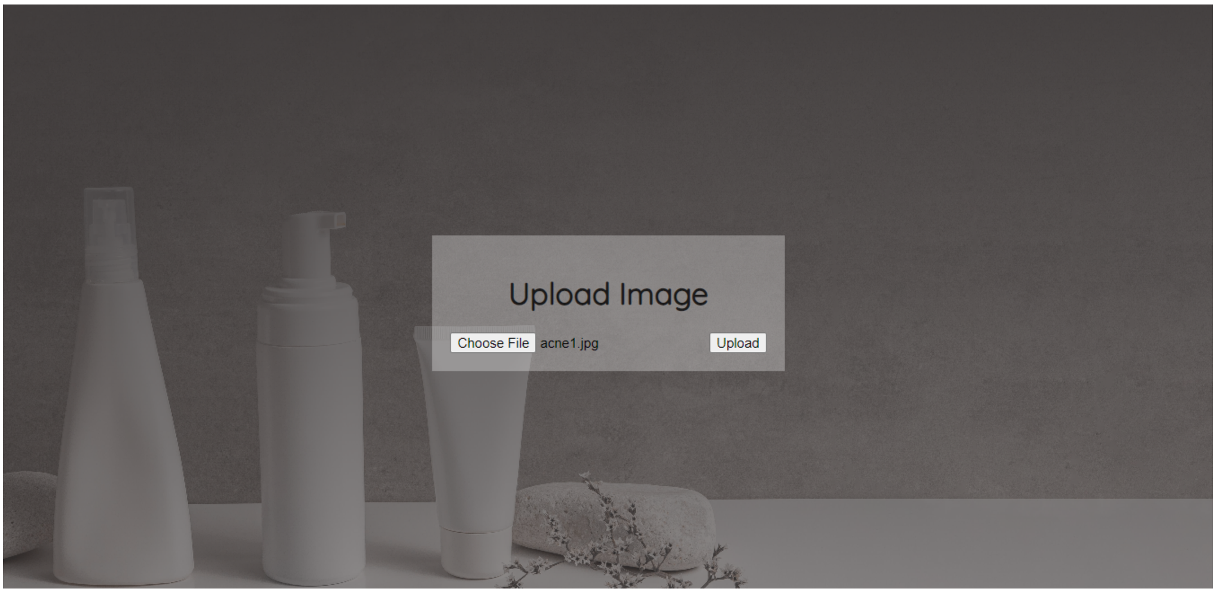
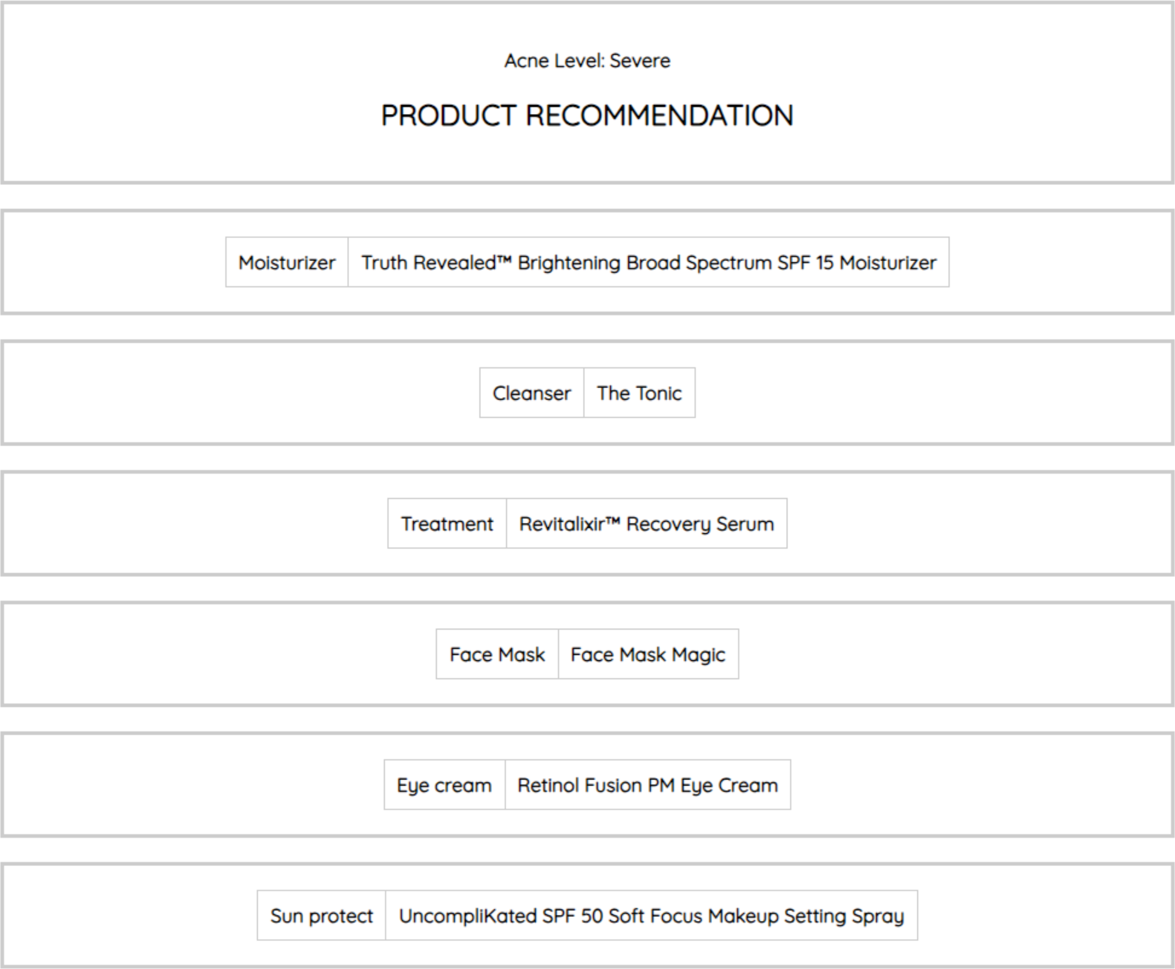


Figure 3.7: Image Upload Page



Acne Level: Severe

PRODUCT RECOMMENDATION

Moisturizer	Truth Revealed™ Brightening Broad Spectrum SPF 15 Moisturizer
Cleanser	The Tonic
Treatment	Revitalixir™ Recovery Serum
Face Mask	Face Mask Magic
Eye cream	Retinol Fusion PM Eye Cream
Sun protect	UncompliKated SPF 50 Soft Focus Makeup Setting Spray

Figure 3.8: Product Recommendation Page

Home Page: In the figure 3.2, home page is the initial page. It provides a get started button that directs the user to the sign in page.

Sign-in Page: In the figure 3.3, sign-in page provides two options: sign-in and create an account. If the user has an account, they are required to enter their email and password and choose sign-in option to move to the skin type options page. If the user does not have an account they have to select the create an account option, which will direct them to the sign-up page.

Sign-up Page: In the figure 3.4, sign-up page provides two options: sign-up and sign-in. The user can create their account by entering their email and password and selecting sign-up option, which will record their details in a database. On selecting sign-in option after submitting their details, they will be directed to the sign-in page.

Skin Type Options Page: In the figure 3.5, skin type options page provides three skin types options: oily skin,dry skin and combination skin, from which the user can choose depending on their skin type. On selecting the option they will be directed to the Image Selection Page/Image Upload Page.

Image Selection Page: In the figure 3.6, Image Selection Page provides an upload file option, where they can choose their facial image.

Image Upload Page: In the figure 3.7, Image upload page , on clicking the upload option the image will get submitted to the image classification model and the user will be directed to the product recommendation page.

Product Recommendation Page: In the figure 3.8, product recommendation page will display the severity of the acne and skin care products based of the skin type selected and the image uploaded by the user.

3.6 Module Division

Data Collection: Implement data collection scripts or workflows to gather relevant data on acne, , skin type and product effectiveness.

Data Preprocessing: Develop modules to preprocess the collected data, handle missing values, remove outliers, and normalize or standardize the data as required. This module ensures the data is clean and in the appropriate format for machine learning algorithms.

Feature Engineering: Create modules to extract or engineer meaningful features from the raw data. This involves transforming and combining existing features or deriving new features that can improve the accuracy of the acne classification and product recommendation models.

Acne Severity Classification: Implement a module for acne severity classification. Evaluate the models using appropriate metrics and select the best-performing model for acne severity classification.

Product Database: Develop a module to create and manage a product database that contains information about various acne treatment products, including their ingredients, effectiveness, user reviews, and other relevant details.

Product Recommendation: Create a module for product recommendation based on acne severity, skin type, and other relevant factors. This module utilizes the product database to suggest suitable products for individual users.

3.7 Algorithm

3.7.1 ADAM Algorithm

Adam Optimizer Adaptive Moment Estimation is an algorithm for optimization technique for gradient descent. The method is really efficient when working with large problem involving a lot of data or parameters. It requires less memory and is efficient. Intuitively, it is a combination of the 'gradient descent with momentum' algorithm and the 'RMSP' algorithm. Adam optimizer involves a combination of two gradient descent methodologies:
1. Momentum: This algorithm is used to accelerate the gradient descent algorithm by taking into consideration the 'exponentially weighted average' of the gradients. Using averages makes the algorithm converge towards the minima in a faster pace.
2. Root Mean Square Propagation (RMSP): Root mean square prop or RMSprop is an adaptive learning algorithm that tries to improve AdaGrad. Instead of taking the cumulative sum of squared gradients like in AdaGrad, it takes the 'exponential moving average'.
Adam Optimizer inherits the strengths or the positive attributes of the above two methods and builds upon them to give a more optimized gradient descent.

3.7.2 Working of the CNN Training

Training a CNN for acne classification involves using a Convolutional Neural Network to identify and classify images based on whether they contain acne or not.

Data Collection and Preparation: Acquiring a labeled dataset containing images of faces or affected skin regions with corresponding labels (acne or non-acne). The dataset should be diverse, balanced, and representative of the problem at hand. Images may need preprocessing (e.g., resizing, normalization) to ensure uniformity.

Architecture Design: Designing the CNN architecture suitable for acne classification. The architecture might consist of several convolutional layers, activation functions like ReLU, pooling layers, and fully connected layers. The exact architecture will depend on factors such as the dataset size and complexity of the classification problem.

Data Augmentation: To prevent overfitting and increase model generalization, data augmentation techniques like rotation, flipping, and translation may be applied to create additional training samples from the original dataset.

Splitting the Dataset: Dividing the dataset into three parts: training set, validation set, and test set. The training set is used for model training, the validation set for hyperparameter tuning, and the test set for final evaluation.

Initialization: Initializing the weights and biases of the CNN. This can be done randomly or using pre-trained weights from a related task (e.g., ImageNet) with transfer learning.

Forward Pass and Loss Calculation: During training, each image from the training set is passed through the CNN layers in a forward direction. The output of the CNN is compared to the true labels of the images using a suitable loss function like binary cross-entropy (since it's a binary classification problem).

Backpropagation and Optimization: The gradients of the loss function with respect to the model's parameters (weights and biases) are calculated using backpropagation. The optimizer (e.g., Adam, SGD) updates the parameters based on these gradients to minimize the loss and improve the model's performance.

Epochs and Batches: The training process is carried out over multiple epochs, where each epoch involves passing the entire training dataset through the network once. To efficiently handle large datasets, training is usually done in batches, where a subset of the training data is processed in each iteration.

Validation and Hyperparameter Tuning: The model's performance is evaluated on the validation set periodically during training. Hyperparameters (e.g., learning rate, dropout rate) may be adjusted based on the validation performance to improve the model's accuracy and generalization.

Stopping Criteria: Training continues until a stopping criterion is met, such as reaching a maximum number of epochs or no significant improvement in the validation accuracy.

Testing: Once training is complete, the final model is evaluated on the test set to assess its performance on unseen data. This step provides an estimate of the model's real-world performance in classifying acne images.

Fine-tuning: Depending on the results, you might consider fine-tuning the model by adjusting the architecture, hyperparameters, or retraining on a combination of the original and additional data. Through this training process, the CNN learns to extract relevant features from acne images and make accurate predictions on new, unseen data, aiding in acne classification.

3.8 Implementation

3.8.1 EfficientNetB0

EfficientNet B0 is a specific architecture in the EfficientNet family of neural networks, which was introduced by Mingxing Tan and Quoc V. Le in their research paper titled ”EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.” EfficientNet aims to provide a scalable and efficient model that achieves state-of-the-art performance on image recognition tasks while being computationally efficient.

The EfficientNet architecture introduces a compound scaling method that uniformly scales the depth, width, and resolution of the neural network. This scaling allows the model to achieve higher accuracy by striking a balance between model size and computational cost.

The notation used in EfficientNet architecture is as follows: EfficientNet Bx: Represents the different model variants in the EfficientNet family, where 'x' denotes a specific model size (e.g., B0, B1, B2, ..., B7).

Depth: The number of layers in the network.

Width: The width of the network, which is a scaling factor applied to the number of channels (filters) in the convolutional layers.

Resolution: The input image resolution.

EfficientNet B0 is the smallest and least computationally intensive variant of the EfficientNet family. It is suitable for resource-constrained environments or when a smaller model is sufficient for the task.

Key characteristics of EfficientNet B0:

Depth: 20 layers

Width: 1.0 (i.e., the number of channels remains the same as the original base model)

Resolution: Typically, the input resolution is 224x224 or 240x240 pixels.

The EfficientNet architecture is built upon a combination of various neural network design techniques, such as:

MobileNetV2-like inverted bottleneck structure: EfficientNet employs depthwise separable convolutions and a bottleneck architecture, similar to MobileNetV2, which helps reduce the computational cost.
Swish Activation Function: EfficientNet uses the Swish activation function, which is a smooth and computationally efficient activation function.
Compound Scaling: EfficientNet applies compound scaling to balance model depth, width, and resolution to achieve optimal performance.

EfficientNet models have demonstrated state-of-the-art performance on various image recognition benchmarks, such as ImageNet, while being more efficient in terms of parameters and computational requirements compared to other large-scale models like ResNet and VGG.

Chapter 4

RESULT AND DISCUSSIONS

The training of our proposal was optimized using the Adam algorithm with the categorical cross-entropy loss function. In addition, all the model parameters were updated during 12 epochs using the early stop method to prevent overfitting.Datasets were used for training and validation. The Kaggle dataset is used to optimize and validate the model, 80% is used for training and 20% for validation.

From 4931 images from the dataset, 3986 for training and 986 for validation purposes. The classification of facial acne involves categorizing acne levels based on their severity, type, and distribution on the face.We define accuracy as the amount of acne with a correct classification. The final trained network achieved 80% accuracy. The classifications in the network were defined in 3 classes: 0-Level 0,1-Level 1,2-Level 2.

Facial acne classification and product recommendation can be achieved using a CNN model used for image classification.Gather a large dataset of facial acne images. This dataset should be labeled with corresponding acne classifications and recommended skincare products. Preprocess the dataset by resizing the images to a consistent size and normalizing the pixel values to enhance model performance. Split the dataset into training and validation sets. The training set will be used to train the CNN model, while the validation set will be used to evaluate its performance and tune hyperparameters.

It typically consists of multiple convolutional layers for feature extraction, followed by fully connected layers for classification. Once the model is trained and evaluated, use it to classify new facial acne images and recommend suitable skincare products. Create a mapping between acne classifications and corresponding skincare products, ensuring that the recommendations align with the specific needs of each acne type and severity.

The following figures (figure 4.1, figure 4.2) shows the model accuracy and model loss respectively:

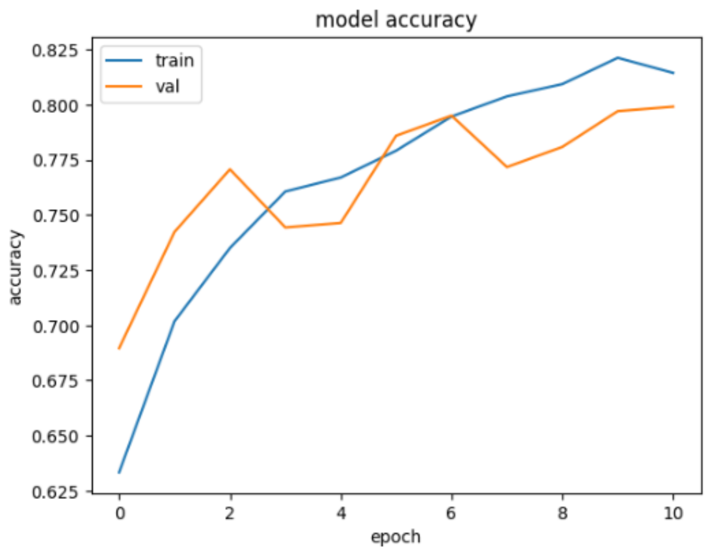


Figure 4.1: Model accuracy while training

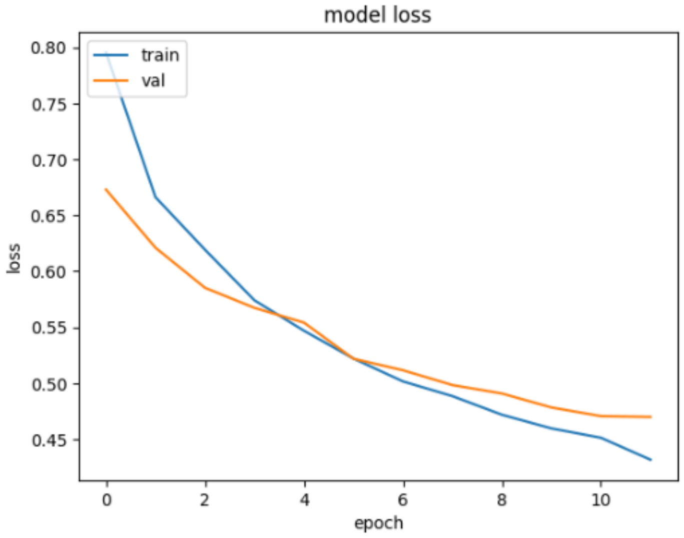


Figure 4.2: Model loss while training

Chapter 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

Facial acne classification plays a crucial role in determining the appropriate treatment and product recommendations for individuals struggling with acne. By understanding the different types of acne and their underlying causes, it becomes easier to select products that target specific concerns and promote clearer skin. Here are some general product recommendations based on common acne classifications:

1. Low Acne: For individuals with occasional breakouts or minor acne, gentle daily cleansing products containing salicylic acid or benzoyl peroxide can be effective in controlling oil production and preventing clogged pores. Non-comedogenic moisturizers and oil-free makeup are also recommended.
2. Moderate Acne: People experiencing moderate acne, characterized by frequent breakouts and inflamed pimples, may benefit from stronger topical treatments such as prescription retinoids or over-the-counter products with higher concentrations of salicylic acid or benzoyl peroxide. Additionally, incorporating a mild exfoliating cleanser or toner with alpha-hydroxy acids (AHAs) can help promote cell turnover and unclog pores.
3. Severe acne: Severe acne is a significant and challenging form of acne that requires proper classification and personalized product recommendations. Machine learning techniques can be employed to tackle this problem by leveraging data and developing models that can accurately classify acne severity and suggest suitable products for treatment.

Everyone's skin is unique, and it's essential to find products that work well for your skin type and concerns. It's recommended to start with a patch test when trying new products and gradually incorporate them into your skincare routine.

5.2 Scope for future work

There is significant potential for future work in facial acne classification and product recommendation. Here are a few areas that hold promise for further research and development:

1. Advanced Imaging Technologies: Advancements in imaging technologies, such as high-resolution photography, 3D imaging, or machine vision, can contribute to more accurate and objective facial acne classification. These technologies can capture detailed information about acne lesions, including their size, shape, color, and texture, enabling better classification algorithms.

2. Artificial Intelligence and Machine Learning: Leveraging artificial intelligence (AI) and machine learning (ML) algorithms can enhance the accuracy and efficiency of facial acne classification. Training models on large datasets with diverse acne images can enable the development of robust classifiers that can identify various types and severity levels of acne with high accuracy.

3. Personalized Product Recommendations: Tailoring product recommendations to individuals' specific acne concerns and skin types is an area that can be further explored. Integrating AI algorithms with personalized skin analysis, including factors like sebum production, skin pH levels, or genetic factors, can help provide targeted product recommendations that are more effective and suitable for individual needs.

4. Integration of Digital Health Technologies: Mobile apps or web-based platforms that allow users to track their acne progression, record skincare routines, and receive personalized product recommendations can be developed. These platforms can also incorporate features such as reminders for medication application, virtual consultations with dermatologists, or community support for individuals with acne.

Overall, future work in facial acne classification and product recommendation should focus on advancing technologies, personalization, and incorporating the latest research to improve acne management outcomes and enhance the overall experience for individuals seeking effective acne solutions.

REFERENCES

[1] Chai, J.; Zeng, H.; Li, A.; Ngai, E. W.T. Deep learning in computer vision: A critical review of emerging techniques and application scenarios. Mach. Learn. Appl. 2021

[2] Hang Zhang ; and Tianyi Ma ; Acne Detection by Ensemble Neural Networks , 2022

[3] Mingxing Tan 1 Quoc V. Le 1; EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, 2020

[4] Amandip Sangha1; Mohammad Rizvi1; Detection of acne by deep learning object detection, 2021

APPENDIX A
CODE

A.1 Training

Import the Libraries

```
import numpy as np
import pandas as pd
import tensorflow as tf
import matplotlib.pyplot as plt
import tensorflow_hub as hub
from tensorflow.keras import layers
```

Split Data set for Training and Validation

```
!pip install split-folders
import splitfolders
splitfolders.ratio('/content/drive/MyDrive/data/Acne', output=”output”,
                   seed=1337, ratio=(.8, 0.2))
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Preprocess data
train_datagen = ImageDataGenerator(rescale=1./255)
valid_datagen = ImageDataGenerator(rescale=1./255)

# Data directories
train_dir = ”/content/output/train/”
test_dir = ”/content/output/val/”

# Import data and create batches
train_data = train_datagen.flow_from_directory(directory=train_dir,
                                               batch_size=16,
                                               target_size=(224, 224),
                                               class_mode="categorical",
                                               seed=42)
valid_data = valid_datagen.flow_from_directory(directory=test_dir,
                                               batch_size=16,
                                               target_size=(224, 224),
                                               class_mode="categorical",
                                               seed=42)
```

Create Facial Acne Classification Model

```
efficientnet_url ="https://tfhub.dev/tensorflow/efficientnet/b0/feature-vector/1"
IMAGE_SHAPE = (224, 224)
BATCH_SIZE = 32

def create_model(model_url, num_classes=3):
  feature_extractor_layer = hub.KerasLayer(model_url,
                                           trainable=False,
                                           name="feature_extraction_layer",
                                           input_shape=IMAGE_SHAPE+(3,))
  model = tf.keras.Sequential([
      feature_extractor_layer,
      layers.Dense(num_classes, activation="softmax", name="output_layer")
  ])
  return model

efficientnet_model = create_model(efficientnet_url, num_classes=3)
```

```
efficientnet_model.compile(loss="categorical_crossentropy",
                     optimizer=tf.keras.optimizers.Adam(),
                     metrics=["accuracy"])
efficientnet_history = efficientnet_model.fit(train_data,
                                  epochs=5,
                                  validation_data=valid_data)
```

Save the Model

```
efficientnet_model.save('saved_model/my_model')
```

A.2 Flask Web App

```

import os
import mysql.connector
from flask import *
import numpy as np
import tensorflow as tf
from werkzeug.utils import secure_filename
from io import BytesIO

from PIL import Image


app = Flask(_name_)
app.static_folder='static'
app.secret_key='secre_key'
class_names = ['Low', 'Moderate', 'Severe']
def get_model():
    global model1
    model1 = tf.keras.models.load_model('saved_model/my_model')
```

```

def load_image(image):
    img = Image.open(image)
    img = img.resize((224, 224))
    img = np.array(img) / 255.0
    img = np.expand_dims(img, axis=0)
    return img

def prediction_acne(img_path):
    new_image = load_image(img_path)
    pred = model1.predict(new_image)
    # print(pred2)
    if len(pred[0]) > 1:
        pred_class = class_names[tf.argmax(pred[0])]
    else:
        pred_class = class_names[int(tf.round(pred[0]))]
    return pred_class

get_model()


@app.route(”/”)
def skin1():
    return render_template(”index.html”)
@app.route(”/Get Started”)
def about():
    return render_template(”signin.html”)

@app.route(”/signin”, methods=['GET', 'POST'])
def signin():
    if request.method == 'POST':
        email = request.form.get('email')
        password = request.form.get('password')

        conn = mysql.connector.connect(
        host='localhost',
        user='root',
        password='1234',
        db='skin'
        )
        cursor = conn.cursor()

        query = "SELECT * FROM customer WHERE email = %s AND password = %s"
        values = (email, password)

        cursor.execute(query, values)
        user = cursor.fetchone()

        cursor.close()
        conn.close()

        if user:
            return redirect('/dashboard')
        else:
            flash('ACCOUNT DOES NOT EXIST','error')

    return render_template('signin.html')

@app.route("/signup", methods=['GET', 'POST'])
def signup():
    if request.method == 'POST':
        email = request.form.get('email')
        password = request.form.get('password')
        conn = mysql.connector.connect(
        host='localhost',
        user='root',
        password='1234',
        db='skin'
        )
        cursor = conn.cursor()

        query = "SELECT * FROM customer WHERE email = %s AND password = %s"
        values = (email,password)

        cursor.execute(query, values)
        user = cursor.fetchone()

        cursor.close()
        conn.close()

        if user:
            flash('ACCOUNT ALREADY EXISTS','error')

        else:
            conn = mysql.connector.connect(
            host='localhost',
            user='root',
            password='1234',
            db='skin'
            )
            cursor = conn.cursor()
            query = "INSERT INTO customer (email, password) VALUES (%s, %s)"
            values = (email, password)

            cursor.execute(query, values)
            conn.commit()

            cursor.close()
            conn.close()
```

```

    return render_template(’signup.html’)

@app.route(”/dashboard”)
def dashboard():
    return render_template(”type.html”)


@app.route(”/Oily Skin”)
def oily():
    global oil, dry1, combi
    dry1= 0
    combi=0
    oil= 1
    return render_template(”upload11.html”)

@app.route(”/Dry Skin”)
def dry():
    global oil, dry1, combi
    combi=0
    oil= 0
    dry1=1
    return render_template(”upload11.html”)

@app.route(”/Combination Skin”)
def comb():
    global oil, dry1, combi
    oil= 0
    dry1=0
    combi=1
    return render_template(”upload11.html”)

@app.route(”/success”, methods = [”POST”])
def success():
    if request.method == ”POST”:
        f = request.files[’file’]
        filename= f.filename
        file_path = os.path.join(’static’,secure_filename(filename))
        f.save(file_path)
        p = prediction_acne(file_path)
        conn = mysql.connector.connect(
        host=’localhost’,
        user=’root’,
        password=’1234’,
        db=’skin’
    )

   # Execute the query to retrieve data based on the prediction
    if p== ”Low”:
      if oil==1:
        cursor = conn.cursor()
        cursor.execute(”SELECT label,name FROM sephore WHERE label=’Moisturizer’ and oily=1 and ingredients like ’%Aloe%’ order by rand() limit 1”)
        data = cursor.fetchall()

        cursor.execute(”SELECT label,name FROM sephore WHERE label=’Cleanser’ and oily=1 and ingredients like ’%Witch%’ order by rand() limit 1”)
        data1 = cursor.fetchall()

        cursor.execute(”SELECT label,name FROM sephore WHERE label=’Treatment’ and oily=1 and ingredients like ’%Retinol%’ order by rand() limit 1”)
        data2 = cursor.fetchall()

        cursor.execute(”SELECT label,name FROM sephore WHERE label=’Face Mask’ and oily=1 and ingredients like ’%Niacin%’ order by
rand() limit 1”)
        data3 = cursor.fetchall()


  cursor.execute(”SELECT label,name FROM sephore WHERE label=’Eye cream’ and oily=1 and ingredients like ’%Retinol%’ order by rand() limit 1”)
        data4 = cursor.fetchall()

        cursor.execute(”SELECT label,name FROM sephore WHERE label=’Sun protect’ and oily=1 and ingredients like ’%Aloe%’ order by rand() limit 1”)
        data5 = cursor.fetchall()


    if dry1==1:
        cursor = conn.cursor()
        cursor.execute(”SELECT label,name FROM sephore WHERE label=’Moisturizer’ and dry=1 and ingredients like ’%Hyaluronic%’ order by rand() limit 1”)
        data = cursor.fetchall()

        cursor.execute(”SELECT label,name FROM sephore WHERE label=’Cleanser’ and dry=1 and ingredients like ’%Glycerin%’ order by rand() limit 1”)
        data1 = cursor.fetchall()

        cursor.execute(”SELECT label,name FROM sephore WHERE label=’Treatment’ and dry=1 and ingredients like ’%Ceramide%’ order by rand() limit 1”)
        data2 = cursor.fetchall()

        cursor.execute(”SELECT label,name FROM sephore WHERE label=’Face Mask’ and dry=1 and ingredients like ’%Shea%’ order by rand()
 limit 1")
            data3 = cursor.fetchall()

            cursor.execute("SELECT label,name FROM sephore WHERE label='Eye cream' and dry=1 and ingredients like '%Oat%' order by rand() limit 1")
            data4 = cursor.fetchall()

            cursor.execute("SELECT label,name FROM sephore WHERE label='Sun protect' and dry=1 and ingredients like '%Aloe%' order by rand() limit 1")
            data5 = cursor.fetchall()


        if combi==1:
            cursor = conn.cursor()
            cursor.execute("SELECT label,name FROM sephore WHERE label='Moisturizer' and combination=1 and ingredients like '%Green Tea%' order by rand() limit 1")
            data = cursor.fetchall()

            cursor.execute("SELECT label,name FROM sephore WHERE label='Cleanser' and combination=1 and ingredients like '%Lactic%' order by rand() limit 1")
            data1 = cursor.fetchall()

            cursor.execute("SELECT label,name FROM sephore WHERE label='Treatment'

and combination=1 and ingredients like '%Niacin%' order by rand() limit 1")
            data2 = cursor.fetchall()

            cursor.execute("SELECT label,name FROM sephore WHERE label='
      Face Mask' and combination=1 and ingredients like '%Jojoba%' order by rand() limit 1")
            data3 = cursor.fetchall()

            cursor.execute("SELECT label,name FROM sephore WHERE label='Eye cream' and combination=1 and ingredients like '%Green Tea%' order by rand() limit 1")
            data4 = cursor.fetchall()

            cursor.execute("SELECT label,name FROM sephore WHERE label='Sun protect' and combination=1 and ingredients like '%Niacin%' order by rand() limit 1")
            data5 = cursor.fetchall()

 elif p== "Moderate":

        if oil==1:
            cursor = conn.cursor()
            cursor.execute("SELECT label,name FROM sephore WHERE label='Moisturizer' and oily=1 and ingredients like '%Niacin%' order by rand() limit 1")
            data = cursor.fetchall()

            cursor.execute("SELECT label,name FROM sephore WHERE label='Cleanser' and oily=1 and ingredients like '%Salicylic%' order by rand() limit 1")
            data1 = cursor.fetchall()

            cursor.execute("SELECT label,name FROM sephore WHERE label='Treatment' and oily=1 and ingredients like '%Benzoyl%' order by rand() limit 1")
            data2 = cursor.fetchall()

            cursor.execute("SELECT label,name FROM sephore WHERE label='
      Face Mask' and oily=1 and ingredients like '%Tea%' order by rand() limit 1")
            data3 = cursor.fetchall()

            cursor.execute("SELECT label,name FROM sephore WHERE label='Eye cream' and oily=1 and ingredients like '%Retinol%' order by rand() limit 1")
            data4 = cursor.fetchall()

            cursor.execute("SELECT label,name FROM sephore WHERE label='Sun protect' and oily=1 and ingredients like '%Aloe%' order by rand() limit 1")
            data5 = cursor.fetchall()

        if dry1==1:
            cursor = conn.cursor()
            cursor.execute("SELECT label,name FROM sephore WHERE label='Moisturizer'

      and dry=1 and ingredients like '%Hyaluronic%' order by rand() limit 1")
            data = cursor.fetchall()

            cursor.execute("SELECT label,name FROM sephore WHERE label='Cleanser' and dry=1 and ingredients like '%Calendula%' order by rand() limit 1")
            data1 = cursor.fetchall()

            cursor.execute("SELECT label,name FROM sephore WHERE label='Treatment' and dry=1 and ingredients like '%Ceramide%' order by rand() limit 1")
            data2 = cursor.fetchall()

            cursor.execute("SELECT label,name FROM sephore WHERE label='
    Face Mask’ and dry=1 and ingredients like ’%Shea%’ order by rand() limit 1”)
            data3 = cursor.fetchall()

            cursor.execute(”SELECT label,name FROM sephore WHERE label=’Eye cream’ and dry=1 and ingredients like ’%Bakuchiol%’ order by rand() limit 1”)
            data4 = cursor.fetchall()

            cursor.execute(”SELECT label,name FROM sephore WHERE label=’Sun protect’ and dry=1 and ingredients like ’%Ceramides%’ order by rand() limit 1”)
            data5 = cursor.fetchall()

        if combi==1:
            cursor = conn.cursor()
            cursor.execute(”SELECT label,name FROM sephore WHERE label=’Moisturizer’ and combination=1 and ingredients like ’%Tea Tree%’ order by rand() limit 1”)
            data = cursor.fetchall()

            cursor.execute(”SELECT label,name FROM sephore WHERE label=’Cleanser’ and combination=1 and ingredients like ’%Salicylic%’ order by rand() limit 1”)
            data1 = cursor.fetchall()

            cursor.execute(”SELECT label,name FROM sephore WHERE label=’Treatment’ and combination=1 and ingredients like ’%Retinol%’ order by rand() limit 1”)
            data2 = cursor.fetchall()

            cursor.execute(”SELECT label,name FROM sephore WHERE label=’Face Mask’ and combination=1 and ingredients like ’%Zinc%’ order by rand() limit 1”)
            data3 = cursor.fetchall()

            cursor.execute(”SELECT label,name FROM sephore WHERE label=’Eye cream’ and combination=1 and ingredients like ’%Retinol%’ order by rand() limit 1”)
            data4 = cursor.fetchall()

            cursor.execute(”SELECT label,name FROM sephore WHERE label=’Sun protect’

    and combination=1 and ingredients like ’%Witch%’ order by rand() limit 1”)
            data5 = cursor.fetchall()



        elif p== ”Severe”:

        if oil==1:
            cursor = conn.cursor()
            cursor.execute(”SELECT label,name FROM sephore WHERE label=’Moisturizer’ and oily=1 and ingredients like ’%Benzoyl%’ order by rand() limit 1”)
            data = cursor.fetchall()

            cursor.execute(”SELECT label,name FROM sephore WHERE label=’Cleanser’ and oily=1 and ingredients like ’%Zinc%’ order by rand() limit 1”)
            data1 = cursor.fetchall()

            cursor.execute(”SELECT label,name FROM sephore WHERE label=’Treatment’ and oily=1 and ingredients like ’%Glycol%’ order by rand() limit 1”)
            data2 = cursor.fetchall()

            cursor.execute(”SELECT label,name FROM sephore WHERE label=’Face Mask’ and oily=1 and ingredients like ’%Aloe%’ order by rand() limit 1”)
            data3 = cursor.fetchall()

            cursor.execute(”SELECT label,name FROM sephore WHERE label=’Eye cream’ and oily=1 and ingredients like ’%Retinol%’ order by rand() limit 1”)
            data4 = cursor.fetchall()

            cursor.execute(”SELECT label,name FROM sephore WHERE label=’Sun protect’ and oily=1 and ingredients like ’%Hyaluronic%’ order by rand() limit 1”)
            data5 = cursor.fetchall()

        if dry1==1:
            cursor = conn.cursor()
            cursor.execute(”SELECT label,name FROM sephore WHERE label=’Moisturizer’ and dry=1 and ingredients like ’%Hyaluronic%’ order by rand() limit 1”)
            data = cursor.fetchall()

            cursor.execute(”SELECT label,name FROM sephore WHERE label=’Cleanser’ and dry=1 and ingredients like ’%Bakuchiol%’ order by rand() limit 1”)
            data1 = cursor.fetchall()

            cursor.execute(”SELECT label,name FROM sephore WHERE label=’Treatment’ and dry=1 and ingredients like ’%Centella%’ order by rand() limit 1”)
            data2 = cursor.fetchall()



```

```
        cursor.execute(”SELECT label,name FROM sephore WHERE label='Face Mask' and dry=1 and ingredients like '%Squalene%' order by rand() limit 1”)
        data3 = cursor.fetchall()

        cursor.execute(”SELECT label,name FROM sephore WHERE label='Eye cream' and dry=1 and ingredients like '%Hyaluronic%' order by rand() limit 1”)
        data4 = cursor.fetchall()

        cursor.execute(”SELECT label,name FROM sephore WHERE label='Sun protect' and dry=1 and ingredients like '%Niacin%' order by rand() limit 1”)
        data5 = cursor.fetchall()


    if combi==1:
        cursor = conn.cursor()
        cursor.execute(”SELECT label,name FROM sephore WHERE label='Moisturizer' and combination=1 and ingredients like '%Licorice%' order by rand() limit 1”)
        data = cursor.fetchall()

        cursor.execute(”SELECT label,name FROM sephore WHERE label='Cleanser' and combination=1 and ingredients like '%Salicylic%' order by rand() limit 1”)
        data1 = cursor.fetchall()

        cursor.execute(”SELECT label,name FROM sephore WHERE label='Treatment' and combination=1 and ingredients like '%Vitamic C%' order by rand() limit 1”)
        data2 = cursor.fetchall()

        cursor.execute(”SELECT label,name FROM sephore WHERE label='
Face Mask’ and combination=1 and ingredients like ’%Retinol%’ order by rand() limit 1”)
        data3 = cursor.fetchall()

        cursor.execute(”SELECT label,name FROM sephore WHERE label=’Eye cream’ and combination=1 and ingredients like ’%Rosemary%’ order by rand() limit 1”)
        data4 = cursor.fetchall()

        cursor.execute(”SELECT label,name FROM sephore WHERE label=’Sun protect’ and combination=1 and ingredients like ’%Titanium%’ order by rand() limit 1”)
        data5 = cursor.fetchall()

    cursor.close()
    conn.close()

    return render_template(’success.html’,p=p,data=data,data1=data1,data2=data2,data3=data3,data4=data4,data5=data5)
       # Close the connection


if _name_ == ”_main_”:
    app.run(port = 12000, debug = True)
```

APPENDIX B

CO-PO AND CO-PSO MAPPING

COURSE OUTCOMES:

After completion of the course the student will be able to

SL. NO	**DESCRIPTION**	**Blooms' Taxonomy Level**
CO1	Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO2	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO4	Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO5	Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)	Level 3: Apply

CO-PO AND CO-PSO MAPPING

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PS O3
C O1	3	3	3	3		2	2	3	2	2	2	3	2	2	2
C O2	3	3	3	3	3	2		3	2	3	2	3	2	2	2
C O3	3	3	3	3	3	2	2	3	2	2	2	3			2
C O4	2	3	2	2	2			3	3	3	2	3	2	2	2
C O5	3	3	3	2	2	2	2	3	2		2	3	2	2	2

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

MAPPING	**LOW/ MEDIUM/ HIGH**	**JUSTIFICATION**
100003/CS622T.1-PO1	**HIGH**	Identify technically and economically feasible problems by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/CS622T.1-PO2	**HIGH**	Identify technically and economically feasible problems by analysing complex engineering problems reaching substantiated conclusions using first principles of mathematics.
100003/CS622T.1-PO3	**HIGH**	Design solutions for complex engineering problems by identifying technically and economically feasible problems.
100003/CS622T.1-PO4	**HIGH**	Identify technically and economically feasible problems by analysis and interpretation of data.
100003/CS622T.1-PO6	**MEDIUM**	Responsibilities relevant to the professional engineering practice by identifying the problem.
100003/CS622T.1-PO7	**MEDIUM**	Identify technically and economically feasible problems by understanding the impact of the professional engineering solutions.
100003/CS622T.1-PO8	**HIGH**	Apply ethical principles and commit to professional ethics to identify technically and economically feasible problems.
100003/CS622T.1-PO9	**MEDIUM**	Identify technically and economically feasible problems by working as a team.
100003/CS622T.1-PO10	**MEDIUM**	Communicate effectively with the engineering community by identifying technically and economically feasible problems.
100003/CS622T.1-P011	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles by selecting the technically and economically feasible problems.
100003/CS622T.1-PO12	**HIGH**	Identify technically and economically feasible problems for long term learning.
100003/CS622T.1-PSO1	**MEDIUM**	Ability to identify, analyze and design solutions to identify technically and economically feasible problems.
100003/CS622T.1-PSO2	**MEDIUM**	By designing algorithms and applying standard practices in software project development and Identifying technically and economically feasible problems.
100003/CS622T.1-PSO3	**MEDIUM**	Fundamentals of computer science in competitive research can be applied to Identify technically and economically feasible problems.
100003/CS622T.2-PO1	**HIGH**	Identify and survey the relevant by applying the knowledge of mathematics, science, engineering fundamentals.

100003/CS622T.2-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems get familiarized with software development processes.
100003/CS622T.2-PO3	**HIGH**	Design solutions for complex engineering problems and design based on the relevant literature.
100003/CS622T.2-PO4	**HIGH**	Use research-based knowledge including design of experiments based on relevant literature.
100003/CS622T.2-PO5	**HIGH**	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes by using modern tools.
100003/CS622T.2-PO6	**MEDIUM**	Create, select, and apply appropriate techniques, resources, by identifying and surveying the relevant literature.
100003/CS622T.2-PO8	**HIGH**	Apply ethical principles and commit to professional ethics based on the relevant literature.
100003/CS622T.2-PO9	**MEDIUM**	Identify and survey the relevant literature as a team.
100003/CS622T.2-PO10	**HIGH**	Identify and survey the relevant literature for a good communication to the engineering fraternity.
100003/CS622T.2-PO11	**MEDIUM**	Identify and survey the relevant literature to demonstrate knowledge and understanding of engineering and management principles.
100003/CS622T.2-PO12	**HIGH**	Identify and survey the relevant literature for independent and lifelong learning.
100003/CS622T.2-PSO1	**MEDIUM**	Design solutions for complex engineering problems by Identifying and survey the relevant literature.
100003/CS622T.2-PSO2	**MEDIUM**	Identify and survey the relevant literature for acquiring programming efficiency by designing algorithms and applying standard practices.
100003/CS622T.2-PSO3	**MEDIUM**	Identify and survey the relevant literature to apply the fundamentals of computer science in competitive research.
100003/CS622T.3-PO1	**HIGH**	Perform requirement analysis, identify design methodologies by using modern tools & advanced programming techniques and by applying the knowledge of mathematics, science, engineering fundamentals.
100003/CS622T.3-PO2	**HIGH**	Identify, formulate, review research literature for requirement analysis, identify design methodologies and develop adaptable & reusable solutions.

100003/CS622T.3-PO3	**HIGH**	Design solutions for complex engineering problems and perform requirement analysis, identify design methodologies.
100003/CS622T.3-PO4	**HIGH**	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
100003/CS622T.3-PO5	**HIGH**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools.
100003/CS622T.3-PO6	**MEDIUM**	Perform requirement analysis, identify design methodologies and assess societal, health, safety, legal, and cultural issues.
100003/CS622T.3-PO7	**MEDIUM**	Understand the impact of the professional engineering solutions in societal and environmental contexts and Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions.
100003/CS622T.3-PO8	**HIGH**	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions by applying ethical principles and commit to professional ethics.
100003/CS622T.3-PO9	**MEDIUM**	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
100003/CS622T.3-PO10	**MEDIUM**	Communicate effectively with the engineering community and with society at large to perform requirement analysis, identify design methodologies.
100003/CS622T.3-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering requirement analysis by identifying design methodologies.
100003/CS622T.3-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by analysis, identify design methodologies and develop adaptable & reusable solutions.
100003/CS622T.3-PSO3	**MEDIUM**	The ability to apply the fundamentals of computer science in competitive research and prior to that perform requirement analysis, identify design methodologies.
100003/CS622T.4-PO1	**MEDIUM**	Prepare technical report and deliver presentation by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/CS622T.4-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems by preparing technical report and deliver presentation.

100003/CS622T.4-PO3	**MEDIUM**	Prepare Design solutions for complex engineering problems and create technical report and deliver presentation.
100003/CS622T.4-PO4	**MEDIUM**	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions and prepare technical report and deliver presentation.
100003/CS622T.4-PO5	**MEDIUM**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and Prepare technical report and deliver presentation.
100003/CS622T.4-PO8	**HIGH**	Prepare technical report and deliver presentation by applying ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
100003/CS622T.4-PO9	**HIGH**	Prepare technical report and deliver presentation effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
100003/CS622T.4-PO10	**HIGH**	Communicate effectively with the engineering community and with society at large by prepare technical report and deliver presentation.
100003/CS622T.4-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles and apply these to one’s own work by prepare technical report and deliver presentation.
100003/CS622T.4-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by prepare technical report and deliver presentation.
100003/CS622T.4-PSO1	**MEDIUM**	Prepare a technical report and deliver presentation to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas.
100003/CS622T.4-PSO2	**MEDIUM**	To acquire programming efficiency by designing algorithms and applying standard practices in software project development and to prepare technical report and deliver presentation.
100003/CS622T.4-PSO3	**MEDIUM**	To apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs by preparing technical report and deliver presentation.
100003/CS622T.5-PO1	**HIGH**	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/CS622T.5-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems by applying engineering and management principles to achieve the goal of the project.

100003/CS622T.5-PO3	**HIGH**	Apply engineering and management principles to achieve the goal of the project and to design solutions for complex engineering problems and design system components or processes that meet the specified needs.
100003/CS622T.5-PO4	**MEDIUM**	Apply engineering and management principles to achieve the goal of the project and use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
100003/CS622T.5-PO5	**MEDIUM**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO6	**MEDIUM**	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities by applying engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO7	**MEDIUM**	Understand the impact of the professional engineering solutions in societal and environmental contexts, and apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO8	**HIGH**	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice and to use the engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO9	**MEDIUM**	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles and apply these to one’s own work, as a member and leader in a team. Manage projects in multidisciplinary environments and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PSO1	**MEDIUM**	The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas. Apply engineering and management principles to achieve the goal of the project.

100003/CS622T.5-PSO2	**MEDIUM**	The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PSO3	**MEDIUM**	The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur and apply engineering and management principles to achieve the goal of the project.