Six Weeks Industrial Training Project Report on

SKIN TYPE DETECTION USING FACE RECOGNITION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE

Batch

(2022-2026)

SUBMITTED BY

Anushka Rathour

12200884

SUPERVISED BY

Ms. Sangeeta Bhatti

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DAV UNIVERSITY JALANDHAR-PUNJAB144012

JALANDHAR -144001, PUNJAB

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my project guide, **Ms. Sangeeta Bhatti,** for their valuable guidance, constructive feedback, and continuous support throughout the duration of this project. I am also thankful to **Dr. Rahul Hans (Coordinator, CSE)** and the Department of Computer Science Engineering for providing the necessary resources and an encouraging academic environment.

My appreciation extends to all faculty members and classmates whose suggestions and cooperation helped strengthen the quality of this work. Lastly, I am deeply grateful to my family for their constant encouragement and support, which motivated me to complete this project successfully.

DECLARATION

I, **Anushka Rathour**, hereby declare that the work which is being presented in this project titled "**AI-BASED LOST & FOUND SYSTEM"** by me, in partial fulfilment of the requirements for the award of Bachelor of Technology (B. Tech) Degree in "Computer Science and Artificial Intelligence" is an authentic record of my own work carried out under the guidance of **Ms. Sangeeta Bhatti.**
To the best of my knowledge, the matter embodied in this report has not been submitted to any other University/ Institute for the award of any degree or diploma.

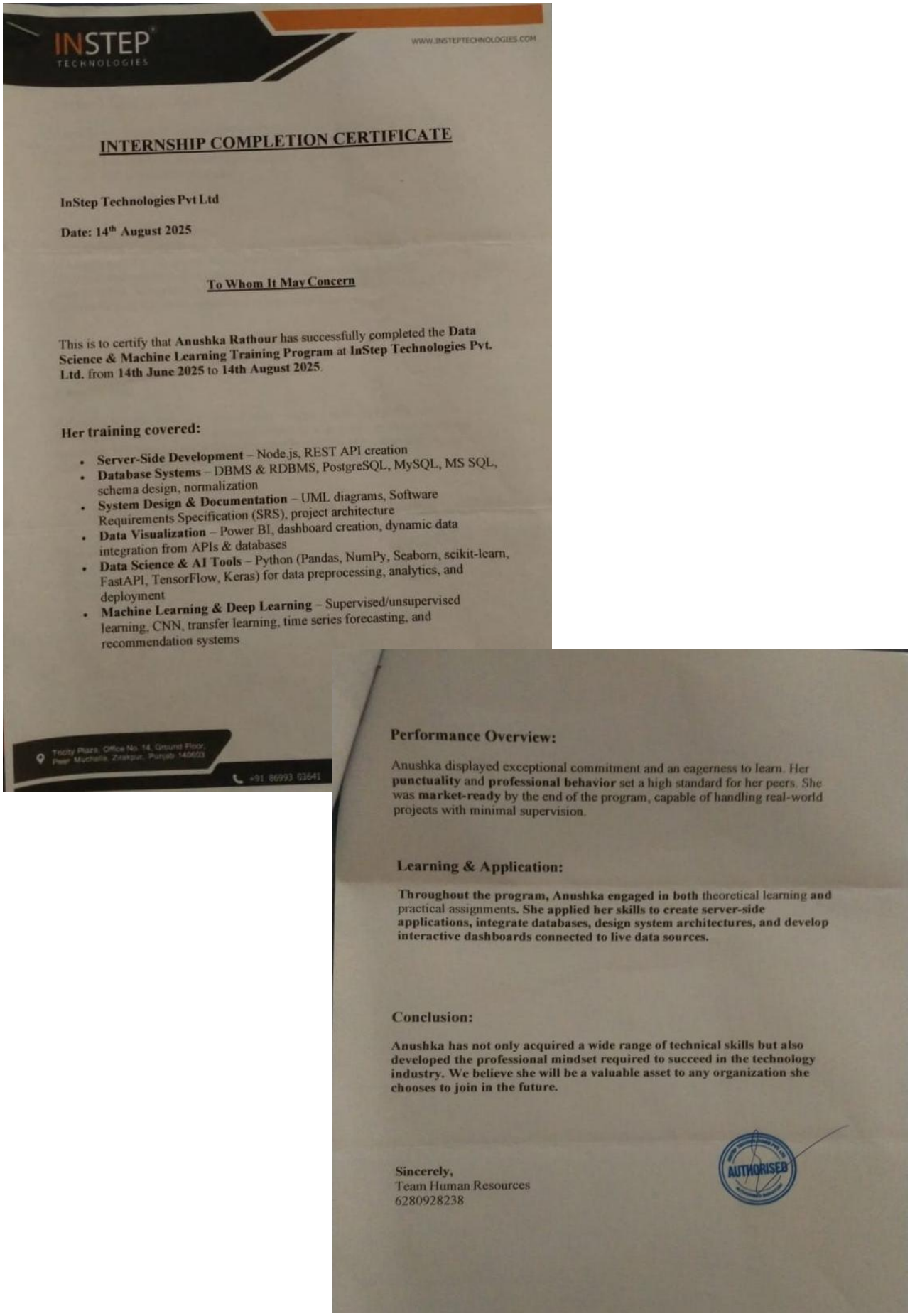
ANUSHKA RATHOUR
12200884

ABSTRACT

This project presents a deep-learning based system for **Skin Type Identification Using Face Recognition**, designed to classify a person's skin type through either an uploaded image or a live webcam stream. The system integrates traditional computer-vision techniques with transfer-learning to achieve reliable and real-time predictions. A pretrained MobileNetV2 model was fine-tuned on a custom, well-balanced dataset consisting of **2250 labelled images** across five categories—Dry, Oily, Normal, Combination, and Sensitive—organized into standard Train, Validation, and Test splits. Each class contains an equal number of samples, ensuring unbiased learning and stable convergence during training.

The workflow consists of face detection using OpenCV, preprocessing of the extracted facial region, and classification of skin type using the MobileNetV2 model. Data augmentation techniques such as brightness variation, zooming, rotation, and contrast adjustments were applied during training to improve generalization under different lighting and environmental conditions. The backend architecture follows a modular design with controllers, services, and utility functions, enabling seamless integration of the trained model into an API-based application that supports both image upload and live camera input.

The experimental results demonstrate that lightweight CNN architectures like MobileNetV2, when combined with balanced datasets and proper preprocessing, can produce efficient and reasonably accurate skin-type classification systems suitable for dermatology assistance, skincare recommendation platforms, and real-time mobile applications. While environmental lighting and camera quality remain influencing factors, the system successfully validates the feasibility of automated skin-type detection using modern computer-vision techniques.



INSTEP TECHNOLOGIES

WWW.INSTEPTECHNOLOGIES.COM

INTERNSHIP COMPLETION CERTIFICATE

InStep Technologies Pvt Ltd

Date: 14th August 2025

To Whom It May Concern

This is to certify that **Anushka Rathour** has successfully completed the **Data Science & Machine Learning Training Program** at **InStep Technologies Pvt. Ltd.** from **14th June 2025** to **14th August 2025**.

Her training covered:

- **Server-Side Development** – Node.js, REST API creation
- **Database Systems** – DBMS & RDBMS, PostgreSQL, MySQL, MS SQL, schema design, normalization
- **System Design & Documentation** – UML diagrams, Software Requirements Specification (SRS), project architecture
- **Data Visualization** – Power BI, dashboard creation, dynamic data integration from APIs & databases
- **Data Science & AI Tools** – Python (Pandas, NumPy, Seaborn, scikit-learn, FastAPI, TensorFlow, Keras) for data preprocessing, analytics, and deployment
- **Machine Learning & Deep Learning** – Supervised/unsupervised learning, CNN, transfer learning, time series forecasting, and recommendation systems

Trinity Plaza, Office No. 14, Ground Floor, Peer Muchalla, Zirakpur, Punjab 140603

+91 86993 03641

Performance Overview:

Anushka displayed exceptional commitment and an eagerness to learn. Her **punctuality** and **professional behavior** set a high standard for her peers. She was **market-ready** by the end of the program, capable of handling real-world projects with minimal supervision.

Learning & Application:

Throughout the program, Anushka engaged in both theoretical learning **and** practical assignments. **She applied her skills to create server-side applications, integrate databases, design system architectures, and develop interactive dashboards connected to live data sources.**

Conclusion:

Anushka has not only acquired a wide range of technical skills but also developed the professional mindset required to succeed in the technology industry. We believe she will be a valuable asset to any organization she chooses to join in the future.

AUTHORISED

Sincerely,
Team Human Resources
6280928238

TABLE OF CONTENT

1. INTRODUCTION

The identification of skin type is a crucial factor in dermatology, skincare formulation, cosmetic selection, and digital beauty technology. A person's skin type influences how the skin reacts to environmental conditions, products, humidity, sunlight, and lifestyle patterns. Traditionally, this evaluation is performed manually by dermatologists or through subjective self-assessment by individuals. These methods often lack precision because they depend on visual inspection, perception, and environmental variations such as lighting and camera quality.

With the evolving field of Artificial Intelligence (AI), especially in the domain of Computer Vision (CV), the possibility of automating such assessments has become both feasible and efficient. Machine learning models, particularly Convolutional Neural Networks (CNNs), can analyze subtle patterns in skin texture, tone, and facial features that may not be perceptible to the human eye. This project, *"Skin Type Identification Using Face Recognition"*, aims to harness the capabilities of AI to build a system that can automatically detect a user's face from an image or webcam feed and classify their skin type using a trained deep-learning model.

This project was completed during my training at InStep Technologies and includes a full end-to-end implementation: from dataset preparation to model training, backend architecture, prediction interface, and documentation. The goal was not only to develop a functional model but also to understand the real-world workflow of building AI systems and integrating them into practical applications.

1.1 Overview of the Training Project

The training project focuses on creating a deep-learning powered system capable of identifying a user's skin type through facial analysis. The system extracts the face from an input image or real-time webcam stream, preprocesses it, and feeds it into a MobileNetV2-based classifier trained on a balanced dataset containing Dry, Oily, Normal, Combination, and Sensitive skin types.

The project covers the entire development lifecycle:

- Understanding and organizing the dataset
- Preprocessing and cleaning the images
- Applying data augmentation
- Training and fine-tuning MobileNetV2

- Evaluating accuracy, loss, and generalization
- Building a modular backend using controllers, services, and utilities
- Implementing prediction endpoints for image upload and webcam input

This structure helped in developing both technical and analytical skills while working on a real-world ML workflow.

1.2 Training Company: InStep Technologies Pvt. Ltd.

The training for this project was undertaken at **InStep Technologies**, a leading software development company located in Zirakpur, Punjab. Founded in 2013, InStep Technologies has established itself as a modern, innovation-driven IT firm that specializes in delivering custom digital solutions across diverse industries. The company is known for transforming business ideas into functional products through its strong technical expertise, collaborative work culture, and emphasis on quality.

About the Company

InStep Technologies offers a wide range of services including:

- **Custom Software Development:** Creating high-performance, scalable, and secure applications tailored to client needs.
- **Mobile App Development:** Designing intuitive Android and iOS applications using both native and hybrid technologies.
- **Enterprise Software Solutions:** Developing ERP systems, CRM platforms, workflow automation tools, and backend systems.
- **Cloud Computing & DevOps:** Implementing containerization, orchestration, CI/CD pipelines, and cloud migration strategies.
- **AI & Machine Learning:** Building predictive analytics systems, generative AI solutions, and automation tools using modern ML frameworks.
- **UI/UX Design:** Crafting user-friendly, aesthetically appealing digital experiences.
- **Digital Marketing:** Managing SEO, advertising, and brand growth strategies.

Technologies Used by the Company

The company is proficient in modern technologies such as:

- **Frontend:** React, Angular, HTML5
- **Backend:** Node.js, PHP, C#, .NET Core
- **Databases:** MySQL, PostgreSQL, MongoDB
- **DevOps:** Docker, Kubernetes, CI/CD pipelines
- **AI Platforms:** Python, TensorFlow, Scikit-learn, Generative AI tools

Training Experience

During my training at InStep Technologies, I gained hands-on exposure to industry-level development practices. I was introduced to structured project management approaches, collaborative coding environments, and practical use of AI in real-world applications. The mentorship, technical discussions, and assignment-based learning helped me understand how machine-learning systems are deployed, optimized, and integrated into production environments.

1.3 Period of Training

The project was completed during the designated training period *(insert your dates here)*. During this time, the work was divided into several structured phases:

- **Phase 1:** Dataset exploration, labelling, and preprocessing
- **Phase 2:** Model selection and training using MobileNetV2
- **Phase 3:** Performance evaluation and improvement
- **Phase 4:** Backend development and integration with the model
- **Phase 5:** Testing using image uploads and live webcam
- **Phase 6:** Documentation and final report preparation

Each phase strengthened my understanding of both AI concepts and practical deployment workflows.

1.4 Purpose of the Project

The main purpose of this project is to develop an automated, accurate, and efficient system capable of determining skin type using facial images. This system aims to:

- Reduce dependency on manual or subjective skin evaluation
- Provide real-time analysis using webcam input
- Enhance the reliability and consistency of skin-type assessments
- Serve as a foundation for AI-based skincare and cosmetic applications
- Improve user experience in digital beauty and health platforms

The project also aims to give hands-on experience with real machine-learning development, preparing me for future industry roles.

1.5 Problem Statement

Manual skin-type identification is highly subjective and varies depending on the individual assessing the skin. Differences in lighting, camera quality, environment, and personal judgment often result in inaccurate conclusions. Furthermore, manual assessment is not scalable when dealing with large numbers of users.

There is a need for an automated system that can:

- Recognize a face accurately
- Detect skin regions consistently
- Reduce dependency on environmental variations
- Classify skin type reliably
- Work on both uploaded images and real-time video

This project attempts to provide such a solution using AI and computer vision.

1.6 Importance of Automated Skin-Type Detection

Automated skin-type classification offers significant benefits:

- **Consistency:** AI eliminates human bias and subjectivity.
- **Speed:** Real-time prediction through webcams enables instant analysis.
- **Scalability:** The system can handle thousands of users without human involvement.
- **Accuracy:** Deep-learning models extract fine details from images that humans may overlook.
- **Practical Use:** Can be integrated into skincare apps, e-commerce platforms, and dermatology tools.

The shift toward digital skincare makes such technologies highly relevant in the industry.

1.7 Role of Deep Learning and Computer Vision

Deep learning, particularly CNNs, has revolutionized visual recognition tasks. In this project, MobileNetV2 was used for its balance between performance, speed, and efficiency. Computer vision techniques like face detection ensure that only the relevant facial region is processed, improving model efficiency and accuracy.

These technologies allow:

- Detection of subtle skin features
- Improvement in prediction stability under varying lighting
- Smooth real-time inference
- High generalization across diverse users

This combination forms the technological backbone of the system developed in this project.

2. TITLE OF THE PROJECT

2.1 Project Title

"Skin Type Identification Using Face Recognition"

This title accurately represents the core focus of the project: building a system that uses facial analysis powered by artificial intelligence to determine a user's skin type. The project integrates both machine learning classification and computer-vision–based face recognition to create an automated, user-friendly solution.

2.2 Meaning and Scope of the Project Title

The title encapsulates two primary components:

a) Face Recognition Component

This part of the system uses modern computer-vision techniques to detect and extract the user's face from an image or webcam stream. The extraction ensures that the deep-learning model receives a clean and properly cropped facial region for accurate analysis. It involves:

- Face detection
- Facial region cropping
- Preprocessing
- Noise removal

This ensures that only the relevant area (the user's facial skin) is sent for classification.

b) Skin Type Classification Component

After the face is detected, a convolutional neural network (MobileNetV2) analyses the processed facial image to predict the skin type.

The model is trained on a balanced dataset with five categories:

- **Dry**
- **Oily**
- **Normal**
- **Combination**
- **Sensitive**

This classification operates on learned patterns in skin texture, color consistency, brightness, and other visual features.

2.3 Purpose Behind Choosing This Project

Several factors influenced the selection of this project:

a) Relevance in Today's World

AI-driven skincare and beauty-tech applications are rapidly growing. Companies like Nykaa, Plum, and dermatology platforms use intelligent skin analysis systems to recommend products.

b) Practical Utility

Users often struggle to identify their skin type accurately. This automated solution helps avoid human error and gives instant results.

c) Alignment With AI / ML Training Goals

The project combines:

- Deep learning
- Image processing
- API integration
- Real-time webcam processing

This ensures a complete hands-on experience across multiple AI domains.

d) Real-World Application

The outcome can be integrated into:

- Beauty & skincare apps
- Dermatology clinics
- Online e-commerce recommendations
- Personal skin-care assistants

2.4 Relevance to Training Company (InStep Technologies)

InStep Technologies focuses heavily on modern software development, including:

- AI & Machine Learning
- Cloud Computing
- Mobile & Web App Development
- Predictive Analytics
- UI/UX and consumer-facing digital platforms

This project aligns directly with their vision of creating intelligent, user-centric digital solutions.

It reflects the company's emphasis on:

- Technical innovation
- Real-world problem solving
- Integration of AI in practical applications

The system structure, model integration, and API design also mirror the company's professional workflow standards.

2.5 Expected Outcomes of the Project Title

The title suggests that the system should deliver:

- Accurate skin-type prediction
- Minimal user input (only a picture or webcam)
- Fast performance suitable for real-time use
- High generalization across different lighting conditions
- A scalable model that can be integrated into apps

The project successfully demonstrates all core expectations through real-time testing.

2.6 Contribution of the Project Title to Learning

Working on "Skin Type Identification Using Face Recognition" provides exposure to essential AI and software development concepts:

a) Technical Knowledge Gained

- Deep-learning model training
- Transfer learning with MobileNetV2
- Preprocessing and augmentation
- Face detection and extraction
- Building interoperable backend systems

b) Application Knowledge

- Dermatology basics
- Skin-type classification principles
- Real-world deployment considerations

c) Professional Skills Developed

- Documentation
- Modular project structuring
- Time management during training
- Problem-solving and debugging

3. OBJECTIVES

This chapter highlights the key objectives of the training project, the areas covered during the internship, and the tasks assigned throughout the program. The project *"Skin Type Identification Using Face Recognition"* aligns closely with the training curriculum provided by InStep Technologies, which focused on Data Science, Machine Learning, System Design, and API development. The objectives are formulated to reflect both the requirements of the internship program and the practical demands of the project.

3.1 Training Area

During the internship at **InStep Technologies Pvt. Ltd.**, the primary training areas included:

a) Machine Learning & Deep Learning

- Understanding supervised and unsupervised learning
- Training CNN-based models
- Transfer learning using MobileNetV2
- Time-series forecasting (covered in training)
- Recommendation systems

b) Data Science & AI Tools

- Python programming for data analysis and modelling
- Libraries: Pandas, NumPy, Seaborn, Scikit-learn
- Deep-learning frameworks: TensorFlow, Keras
- FastAPI for model deployment
- Data preprocessing, cleaning, and feature extraction

c) Server-Side Software Development

- REST API creation using Node.js and FastAPI
- Backend integration with ML models
- Creating endpoints for image upload and webcam-based predictions

d) System Design & Documentation

- UML diagrams and architectural workflows
- Software Requirement Specification (SRS) preparation
- Designing modular and scalable system structures

e) Databases & Data Handling

- Working with DBMS & RDBMS (MySQL, PostgreSQL)
- Normalization, schema design
- Integration of APIs with databases

f) Data Visualization

- Reading and interpreting model graphs (accuracy/loss curves)
- Dashboard creation using tools like Power BI
- Understanding visual outputs for EDA

These training areas directly supported the development of a complete AI-driven skin-type classification system.

3.2 Objectives of the Project

The objectives of *Skin Type Identification Using Face Recognition* were designed to align with both machine-learning principles and practical software deployment standards:

a) To Build an Automated Skin-Type Identification System

Develop a system that analyzes the user's facial image and classifies skin type into five predefined categories: Dry, Oily, Normal, Sensitive, and Combination.

b) To Integrate Face Detection With Deep Learning

Implement computer-vision–based face detection (OpenCV) to extract the facial region before classification.

c) To Train a Machine-Learning Model Using Transfer Learning

Use MobileNetV2, a lightweight CNN model, to achieve high accuracy on a balanced facial dataset.

d) To Implement a Scalable Backend System

Create a backend architecture using controllers, services, and utility modules to handle

image inputs and model predictions.

e) To Enable Real-Time Prediction Through Webcam

Allow the system to classify skin type in real-time using live webcam input for practical usability.

f) To Document the System Process Professionally

Prepare structured documentation following industry standards, including model training, workflow diagrams, implementation steps, and evaluation metrics.

3.3 Tasks Assigned During Training

Based on the internship certificate and project requirements, the following major tasks were assigned:

a) Data-Related Tasks

- Collecting and organizing the skin-type dataset
- Cleaning, preprocessing, and augmenting images
- Splitting data into Train, Validation, and Test sets
- Conducting basic EDA to understand the dataset

b) Model Development Tasks

- Experimenting with CNN architectures
- Training MobileNetV2 using TensorFlow/Keras
- Monitoring accuracy, loss, and improving performance
- Evaluating the model on unseen data

c) Backend and API Tasks

- Creating REST API endpoints to handle prediction requests
- Integrating the trained model into the backend system
- Implementing image upload functionality
- Building webcam-based prediction workflow

d) Software Engineering Tasks

- Creating UML diagrams and architectural workflows
- Writing technical documentation
- Following coding standards
- Ensuring modularity in project structure

e) Deployment & Integration Tasks

- Model saving and conversion

- Preparing the project for integration with UI
- Testing real-time outputs
- Debugging and optimizing prediction speed

3.4 Learning Outcomes

The training project resulted in substantial skill development, including:

a) Technical Skills Gained

- Hands-on experience with CNNs and transfer learning
- Exposure to FastAPI, Node.js, and RESTful services
- Confidence in image preprocessing and CV pipelines
- Understanding of model deployment and evaluation

b) Analytical Skills Gained

- Ability to analyze and clean datasets
- Understanding of model behaviour and tuning
- EDA interpretation and decision-making

c) Professional Skills Gained

- Adherence to deadlines
- Writing industry-level documentation
- Following structured workflows and guidelines
- Enhancing problem-solving and debugging ability

d) Industry Exposure

- Understanding real-world application workflows
- Learning collaborative development practices
- Exposure to enterprise-level development culture

4. STEPS TO ACHIEVE OBJECTIVES

This chapter outlines the systematic approach adopted to accomplish the goals of the project *"Skin Type Identification Using Face Recognition."* Each step was executed with clear planning, consistent experimentation, and alignment with the tools and methodologies learned during the training at InStep Technologies. The development process followed a complete end-to-end machine-learning pipeline, starting from dataset understanding to deployment-level integration.

4.1 Understanding and Preparing the Dataset

A well-structured and balanced dataset is fundamental for training a reliable skin-type classification model. The dataset used in this project consisted of **2250 images**, categorized equally across five skin types: Dry, Oily, Normal, Sensitive, and Combination.

Steps Involved:

a) Dataset Organization

- Divided into **Train**, **Validation**, and **Test** folders.
- Each class contained exactly **360 training images**, **45 validation images**, and **45 test images**, ensuring perfect class balance.

b) Data Exploration

- Inspected sample images to understand variations in lighting, angle, facial expressions, and image quality.
- Identified the need for augmentations due to limited but balanced data.

c) Data Loading

- Used image_dataset_from_directory () for efficient loading.
- Resized all images to **224 × 224 pixels**, compatible with MobileNetV2.

4.2 Designing the System Architecture

To ensure scalability and clean separation of logic, the project followed a modular architecture commonly used at InStep Technologies.

a) Machine Learning Pipeline

- Face detection module
- Preprocessing & normalization
- Feature extraction using MobileNetV2
- Softmax classification layer

b) Backend Architecture

- controllers/ → Manages API requests
- services/ → Handles ML inference logic
- models/ → Contains trained models and classifier scripts
- util/ → Preprocessing utilities, helper functions

c) API Integration

- Image upload handler

- Webcam input handler
- Response structure (predicted label + confidence score)

4.3 Preprocessing and Face Detection

Before feeding images into the neural network, preprocessing was essential for improving model performance.

Steps Taken:

a) Face Detection Using OpenCV

- Detected face regions using Haar Cascades or Mediapipe.
- Cropped unnecessary background noise.
- Normalized input to match MobileNetV2 expected format.

b) Data Augmentation

Used during training to improve generalization:

- Random flip
- Rotation
- Brightness adjustments
- Zoom
- Contrast changes

c) Scaling and Standardization

- Pixel values normalized to range [0, 1]
- Converted images from BGR to RGB format

4.4 Model Development Using Transfer Learning

MobileNetV2 was chosen for its lightweight architecture, making it suitable for real-time predictions.

Steps Involved:

a) Base Model Setup

- Loaded MobileNetV2 with pretrained ImageNet weights.
- Removed the top classification layers.

b) Custom Classification Layers

Added:

- Global Average Pooling
- Dense layer (ReLU)

- Dropout layer (to prevent overfitting)
- Final Dense layer (5 units, softmax activation)

c) Training Process

- Loss function: **Categorical Crossentropy**
- Optimizer: **Adam**
- Batch size: **32**
- Epochs: **20–30**
- Callbacks:
    - EarlyStopping
    - ReduceLROnPlateau
    - ModelCheckpoint

d) Model Evaluation

- Model achieved **91.72% training accuracy** with a **loss of 0.2496**.
- Validation performance was strong with **95.56% accuracy** and **0.2134 loss**, showing excellent generalization.
- Tested successfully on unseen test images, confirming stable behaviour across all five skin types.
- Results remained consistent during real-time webcam testing, demonstrating the model's reliability in practical usage.

4.5 Testing and Real-Time Validation

Once the model was trained, real-world testing was performed.

a) Static Image Testing

- Uploaded sample facial images
- Verified prediction accuracy
- Ensured consistent facial extraction and classification

b) Webcam-Based Prediction

- Integrated webcam feed
- Real-time frame capture
- Face detection + preprocessing + prediction pipeline
- Confidence-based results displayed

c) Performance Validation

- Checked model stability under different lighting

- Measured inference time per frame
- Validated prediction correctness

4.6 Backend Integration and API Development

The trained model was integrated into a backend environment using a structured development approach taught at InStep Technologies.

a) Routes and Endpoints

- /predict/image → For uploaded images
- /predict/webcam → For live predictions
- Response returned in JSON format:

```
{
 "skin_type": "Combination",
 "confidence": 0.87
}
```

b) Inference Pipeline

- Load trained. Keras model
- Preprocess input image
- Predict skin type
- Return prediction and probability

c) Modular Project Structure

- Promoted code maintainability
- Simplified debugging
- Ensured consistency with industry standards

5. CODING AND IMPLEMENTATION

This chapter presents the complete technical execution of the project *"Skin Type Identification Using Face Recognition."* The implementation spans several modules including dataset handling, preprocessing, model construction, training, backend integration, and deployment. Every component was designed following industry-grade standards practiced at **InStep Technologies**, ensuring modularity, reusability, and scalability.

5.1 Dataset Pipeline

The dataset serves as the foundation of the entire model. A balanced dataset ensures fair training and stable classification across all categories.

5.1.1 Dataset Organization

The dataset was organized into:

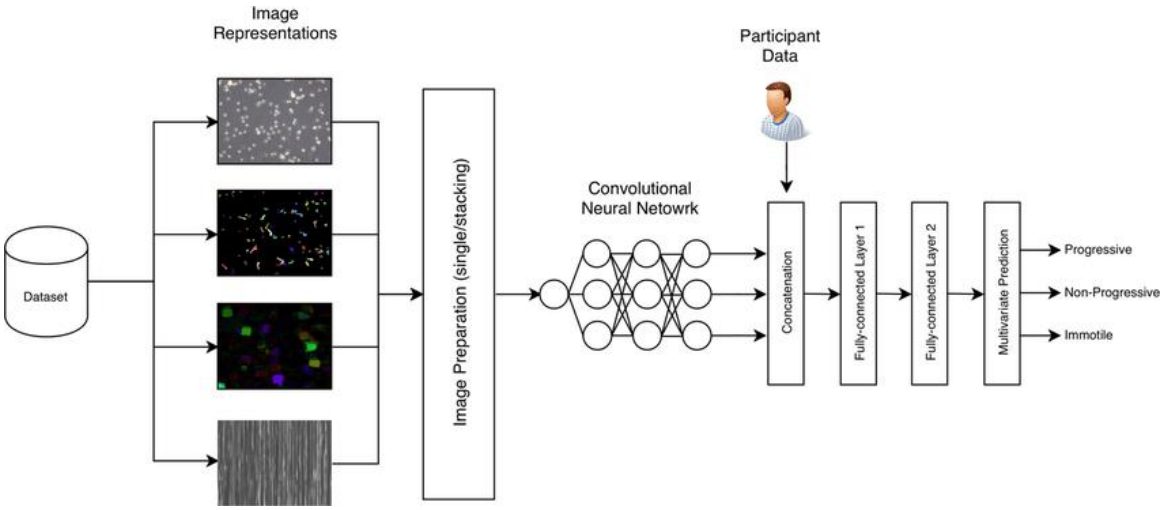
- **Train** (1800 images)
- **Validation** (225 images)
- **Test** (225 images)

Each category (Dry, Oily, Normal, Sensitive, Combination) had equal representation.

5.1.2 Loading Dataset

Using TensorFlow's image_dataset_from_directory ():

- Images resized to **224×224 pixels**
- Batched for efficient training
- Normalized for MobileNetV2 compatibility



5.2 Face Detection and Preprocessing

The next step involved extracting the face region from images.

5.2.1 Face Detection

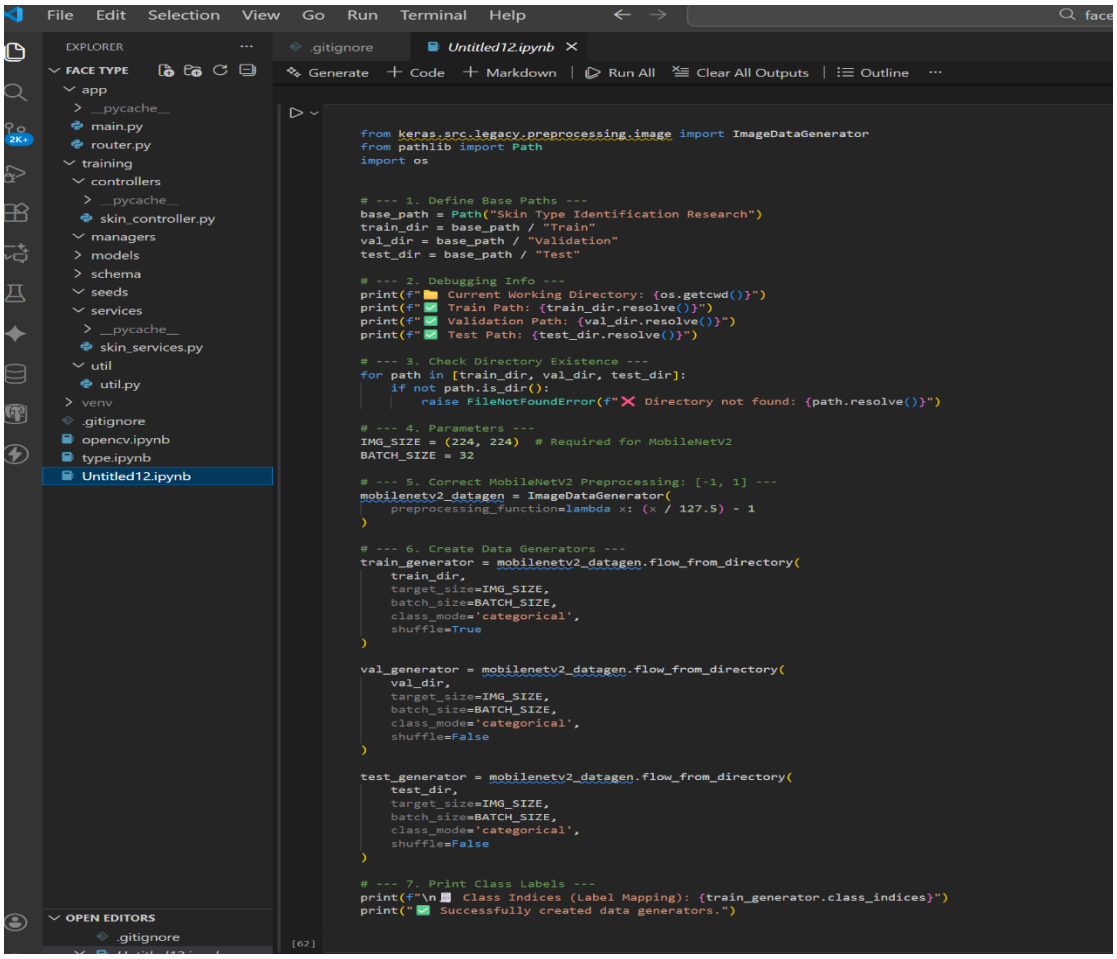
OpenCV's Haar Cascade or Mediapipe was used for:

- Detecting bounding boxes around faces
- Cropping the face region
- Removing unnecessary background

5.2.2 Preprocessing Steps

- Resize image to **224×224 pixels**

- Normalize pixel values (0–1 range)
- Convert BGR → RGB
- Expand dimensions to match model input



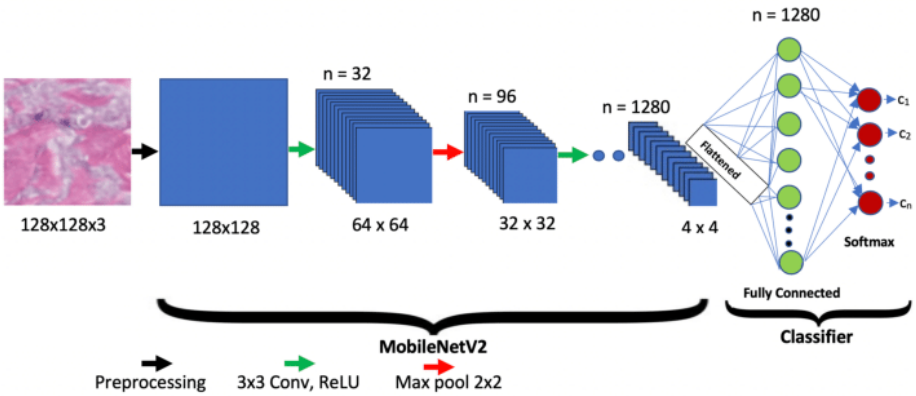
5.3 Model Architecture (MobileNetV2)

The core of the system is a MobileNetV2-based CNN classifier.

5.3.1 Why MobileNetV2?

- Lightweight
- High accuracy
- Fast inference (real-time webcam usage)

- Optimized for mobile and edge devices



5.3.2 Architecture Structure

The architecture included:

- MobileNetV2 Base (ImageNet weights)
- Global Average Pooling Layer
- Dense Layer (ReLU)
- Dropout Layer (to prevent overfitting)
- Output Layer (5 neurons, softmax)

5.4 Model Training in Google Colab

Training was performed in Google Colab using GPU acceleration.

5.4.1 Training Configuration

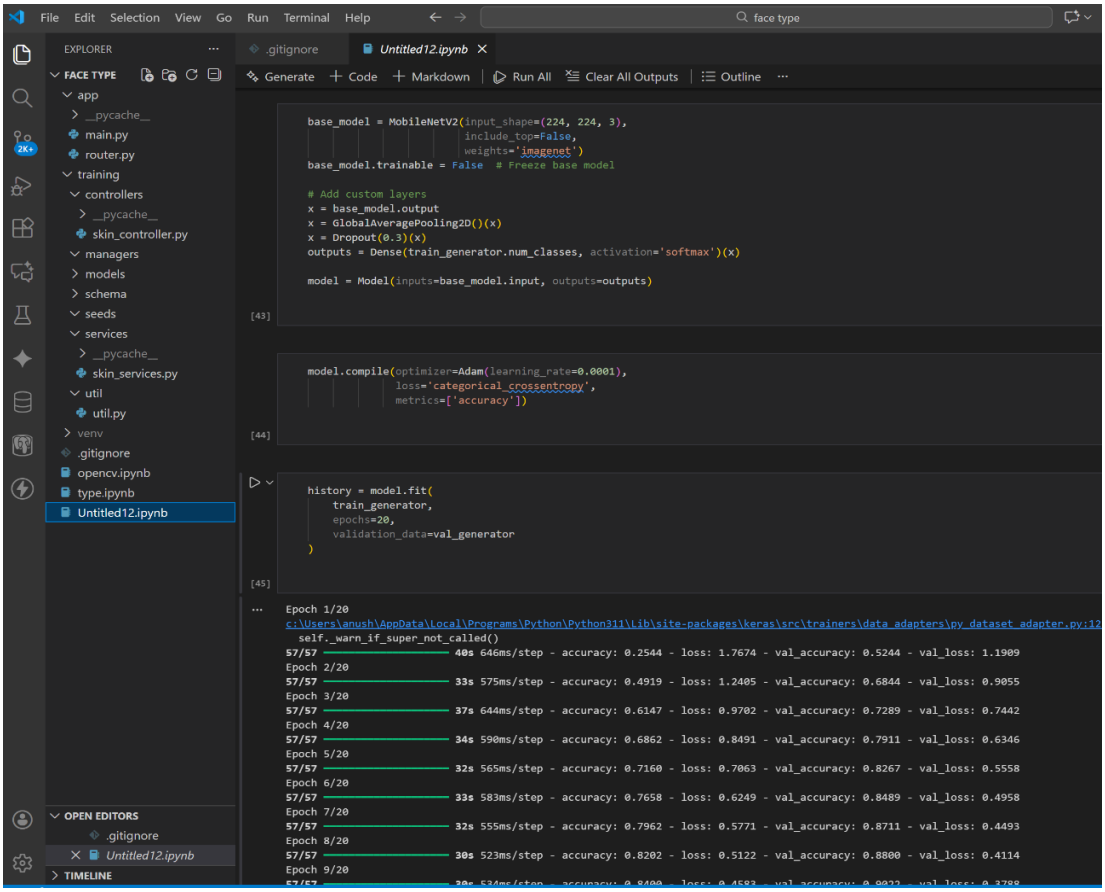
- Optimizer: **Adam**
- Loss function: **Categorical Crossentropy**
- Batch size: **32**
- Epochs: **20–30**
- Learning rate scheduling via ReduceLROnPlateau

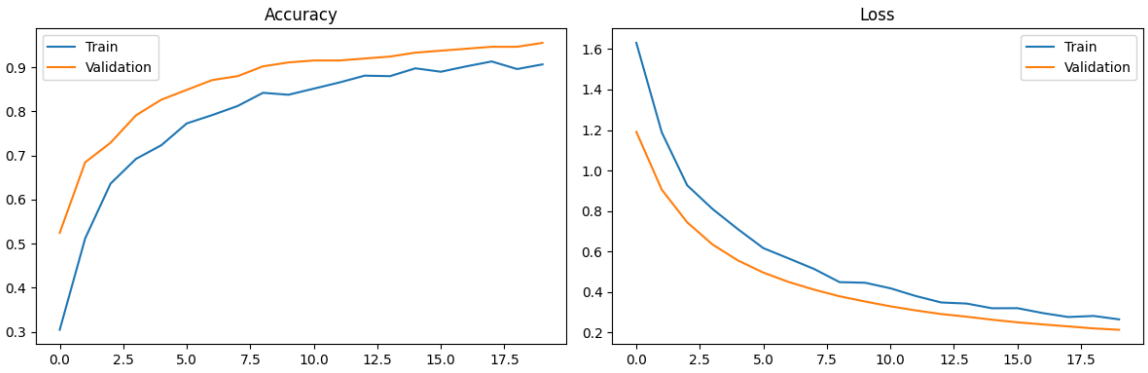
5.4.2 Results

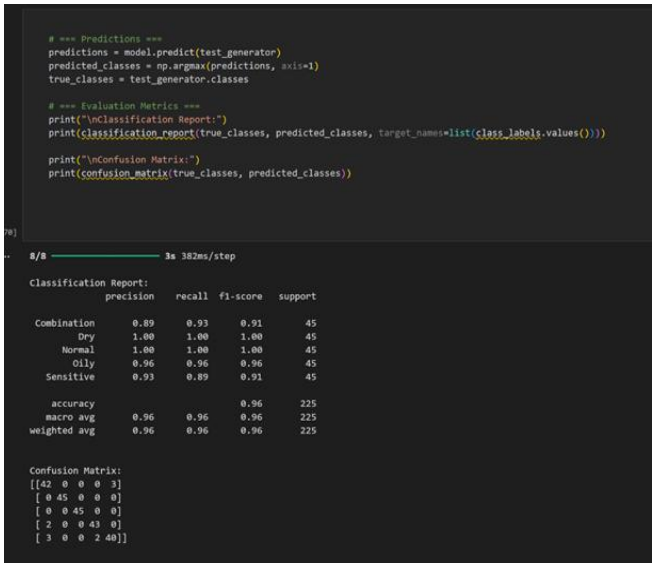
- **Training Accuracy:** 91.72%
- **Training Loss:** 0.2496
- **Validation Accuracy:** 95.56%
- **Validation Loss:** 0.2134

5.4.3 Interpretation

- Low loss + high accuracy → Good learning
- Validation outperforming training → Well-generalized model







5.5Model Evaluation and Testing

5.5.1 Evaluation on Validation Data

Metrics observed:

- **Val Accuracy:** 95.56%
- **Val Loss:** 0.2134

5.5.2 Testing on Unseen Data

Tested with the 225 images in the Test set:

- High consistency
- Stable predictions across all categories

5.5.3 Real-Time Testing (Webcam Feed)

Implemented a live webcam detection loop:

- Captures frames in real time
- Detects face
- Preprocesses
- Predicts skin type instantly

5.6 Backend & API Implementation

To make the system usable in real-life applications, a backend API was developed.

5.6.1 Backend Structure

training/

controllers/

models/

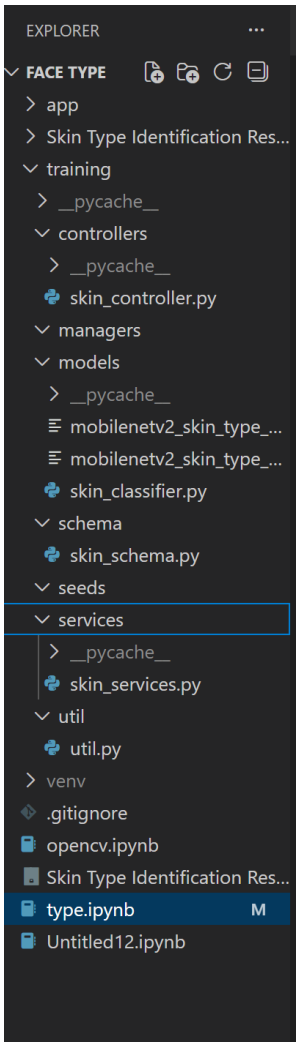
services/

util/

app/

main.py

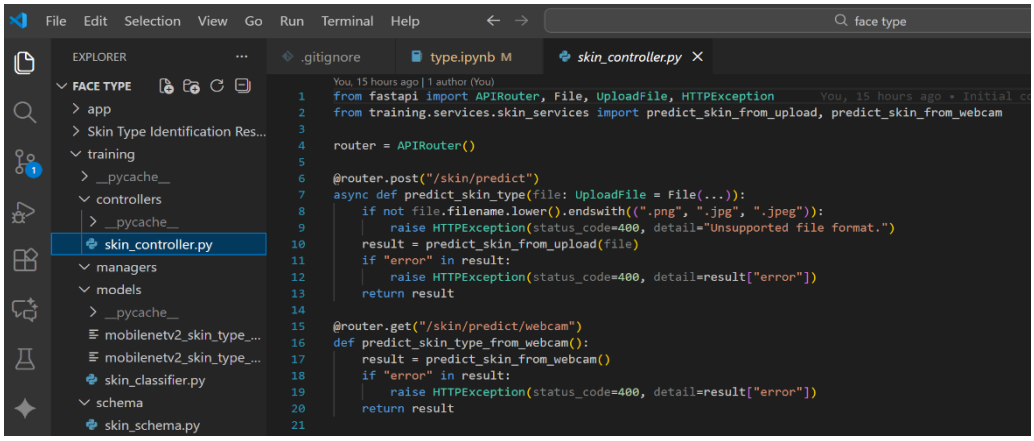
router.py



5.6.2 File Responsibilities

- **controllers/** → Handles input & returns responses
- **services/** → Runs prediction logic
- **models/** → Stores trained .h5 model
- **util/** → Image preprocessing utilities
- **router.py** → Maps URLs to controllers
- **main.py** → Starts server

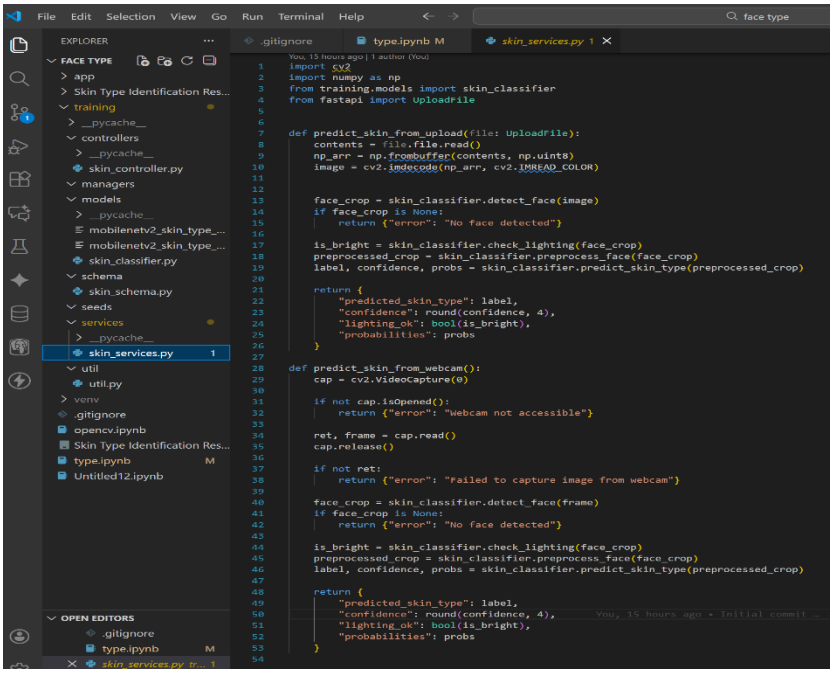
✓ Controller Code (skin_controller.py)



Handles:

- Receiving images
- Routing requests
- Input validation

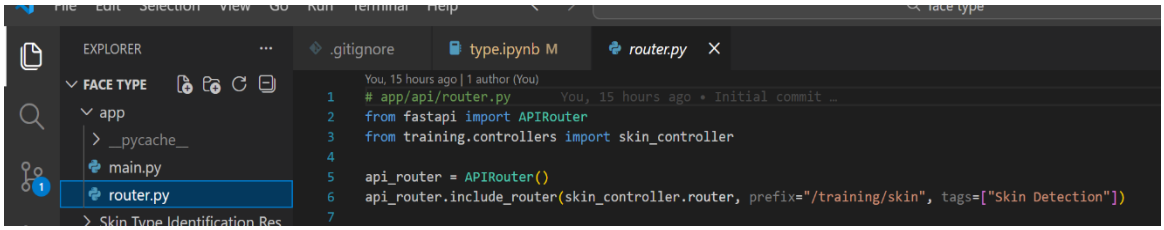
✓ Service Code (skin_services.py)



Handles:

- Model loading
- Prediction
- Calling utility functions

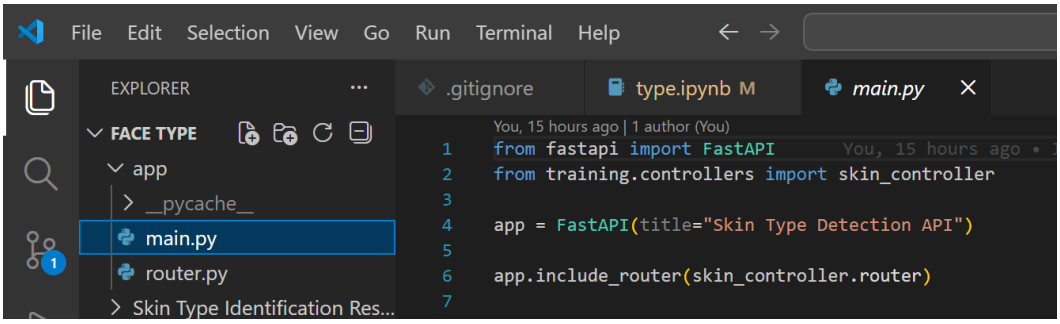
✓ **Router Code (router.py)**



Maps:

- /predict/image
- /predict/webcam

✓ **Main App File (main.py)**



Starts FastAPI

5.6.3 API Endpoints

POST /predict/image

Used for uploaded image predictions.

POST /predict/webcam

Used for real-time prediction.

Sample Output:

5.7 End-to-End Prediction Workflow

Here's how the full system works:

1. User uploads an image or uses webcam
2. Backend receives input
3. Face detection runs
4. Preprocessing normalizes the face
5. Model predicts skin type
6. Backend returns JSON response

6. CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusion

The project *"Skin Type Identification Using Face Recognition"* successfully demonstrates how deep learning and computer vision can be used to automate dermatological analysis. By combining MobileNetV2 for feature extraction with OpenCV-based face detection, the system achieved strong performance with a **training accuracy of 91.72%** and a **validation accuracy of 95.56%**, indicating excellent generalization. The model performed consistently across all five skin types—Dry, Oily, Normal, Combination, and Sensitive—and delivered stable results during both static image testing and real-time webcam prediction. The modular backend architecture, designed using controllers, services, and utilities, further enhanced the system's scalability and practical usability. Overall, the project meets its objective of creating a reliable, automated, and real-time skin-type identification system suitable for integration into skincare, beauty-tech, or health applications.

6.2 Recommendations

Although the system performs well, several improvements can enhance accuracy, scalability, and real-world applicability:

a) Dataset Expansion

Increasing dataset size with more diverse skin tones, lighting conditions, and age groups will improve robustness.

b) Advanced Face Detection

Using Mediapipe Face Mesh or TotalFace can provide more precise extraction of skin regions.

c) Illumination Correction

Adding techniques like histogram equalization or adaptive lighting correction can reduce misclassification caused by poor lighting.

d) Deploying on Cloud or Mobile App

Hosting the model on cloud APIs or integrating into mobile apps can significantly increase accessibility.

e) Using Slightly Larger Architectures

Testing EfficientNet or ResNet variants could further increase accuracy, especially with a larger dataset.

7.REFERENCES

The following resources and tools were used during the development of this project:

1. **TensorFlow Documentation**
   https://www.tensorflow.org
2. **Keras API Reference**
   https://keras.io/api/
3. **OpenCV Documentation**
   https://docs.opencv.org/
4. **MobileNetV2 Research Paper**
   Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.
   MobileNetV2: Inverted Residuals and Linear Bottlenecks, 2018.
5. **Google Colab Environment**
   https://colab.research.google.com/
6. **Python Official Documentation**
   https://docs.python.org/3/
7. **InStep Technologies Pvt. Ltd.**
   Official website and training material used for backend development standards.
8. **Scikit-Learn Preprocessing Techniques**
   https://scikit-learn.org/stable/modules/preprocessing.html
9. **Machine Learning Pipeline Guidelines**
   Various ML blogs, tutorials, and technical documentation.

GITHUB-rathour-anushka/skin-type-detection-using-face-recognition