A PROJECT SYNOPSIS

On

Face-Fit AI: Virtual Try-On

Submitted By

1. **SHRUTI PEDNEKAR (55)**
2. **CHITRANG SAWANT (62)**
3. **LABH SAWANT (63)**
4. **SNEHA SHINDE (66)**

Under the Guidance of

Prof. Amita Raman

Department of Computer Science and Engineering In

Artificial Intelligence and Machine Learning

Saraswati Education Society's

SARASWATI COLLEGE OF ENGINEERING

Kharghar,Navi Mumbai(Affiliated to University of Mumbai)

Saraswati College of Engineering, Kharghar

Academic Year :-2025-26

Vision

"To be globally recognized as an autonomous institute in engineering education with an emphasis on academics, research and skills enhancement to create innovators and future leaders for Industry and Societal needs."

Mission

M1: To provide theoretical and practical knowledge through quality teaching learning to develop competent engineers.

M2: To create an ambience that facilitates research, entrepreneurship and collaborations leading to emergence of innovators and future leaders.

M3: To develop a student-centric approach that inculcates moral, ethical values and leadership skills for holistic development.

Department of Computer Science and Engineering In

Artificial Intelligence and Machine Learning

Vision

"To achieve an excellent standard of quality education in the field of AI &ML
through the right academic ambience and research environment to create Innovators & Future leaders to serve Industry & Society."

Mission

1. To establish a solid foundation in AI & ML through quality teaching learning in theoretical & practical aspects to inculcate innovative mindset in students.
2. To promote excellence by encouraging Entrepreneurship & Research and to acquaint students with recent trends in Industry.
3. To impart value based education to contribute positively in holistic development of students with moral, ethical and leadership skills.

SARASWATI Education Society's

SARASWATI College of Engineering

Learn Live Achieve and Contribute

Kharghar, Navi Mumbai - 410 210.

Recognised by DTE, Govt. of Maharashtra, Approved by AICTE, Affiliated to University of Mumbai

AUTONOMOUS | NAAC A+ | NBA (2 - UG PROGRAMS)

CERTIFICATE

*This is to certify that the requirements for the synopsis entitled, "**Face-Fit AI: Virtual Try On**" have been successfully completed by the following students:*

Roll numbers	Name
55	**SHRUTI PEDNEKAR**
62	**CHITRANG SAWANT**
63	**LABH SAWANT**
66	**SNEHA SHINDE**

In partial fulfillment of Sem V **Bachelor of Engineering of Mumbai University, CSE-AIML** of Saraswati college of Engineering, Kharghar during the academic year 2025-26.

Internal Guide | **External Examiner**

Prof. Amita Raman

Project coordinator | **Head of Department**

Prof. Amita Raman | Prof. Anuja Chandane

FACE-FIT AI: VIRTUAL TRY-ON

ABSTRACT

The modern pursuit of personal style is often hindered by a lack of objective guidance tailored to an individual's unique physical attributes. The "FaceFit" project addresses this challenge by developing an intelligent system that provides personalized style recommendations based on facial analysis. This project leverages a multi-faceted approach, integrating computer vision for facial feature detection, machine learning for classification, and generative AI for creating nuanced style guides. The system processes a user-provided facial image to accurately identify 68 key facial landmarks using the Dlib library. These landmarks are then translated into a quantitative feature vector, which is fed into a pre-trained RandomForestClassifier to determine the user's face shape (e.g., Oval, Round, Square) and gender. Subsequently, this classification data is used to query both a local database and the Google Gemini API to generate tailored recommendations for hairstyles, eyewear, and color palettes. The final output is presented to the user through a clean, intuitive graphical user interface (GUI) built with Tkinter. The project successfully demonstrates a practical application of AI in the domain of personal styling, offering users an accessible and data-driven tool for enhancing their appearance.

INDEX

Program Educational Objectives (PEO)

1. To apply statistical data analysis, Artificial Intelligence and Machine Learning techniques to effectively solve real-world problems.
2. To motivate & prepare students for lifelong learning and research to manifest global competitiveness.
3. To equip students with communication, teamwork and leadership skills to accept challenges in all facets of life ethically.

Program Outcomes (PO)

PO 1: Engineering knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop the solution of complex engineering problems.

PO 2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)

PO 3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)

PO 4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8)

PO 5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)

PO 6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).

PO 7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)

PO 8: Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.(WK9)

PO 9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences

PO 10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

PO 11: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8)

Knowledge and Attitude Profile (WK)

WK1: A systematic, theory-based understanding of the natural sciences applicable to the discipline and awareness of relevant social sciences.

WK2: Conceptually-based mathematics, numerical analysis, data analysis, statistics and formal aspects of computer and information science to support detailed analysis and modelling applicable to the discipline.

WK3: A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline.

WK4: Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline; much is at the forefront of the discipline.

WK5: Knowledge, including efficient resource use, environmental impacts, whole-life cost, reuse of resources, net zero carbon, and similar concepts, that supports engineering design and operations in a practice area.

WK6: Knowledge of engineering practice (technology) in the practice areas in the engineering discipline.

WK7: Knowledge of the role of engineering in society and identified issues in engineering practice in the discipline, such as the professional responsibility of an engineer to public safety and sustainable development.

WK8: Engagement with selected knowledge in the current research literature of the discipline, awareness of the power of critical thinking and creative approaches to evaluate emerging issues.

WK9: Ethics, inclusive behavior and conduct. Knowledge of professional ethics, responsibilities, and norms of engineering practice. Awareness of the need for diversity by reason of ethnicity, gender, age, physical ability etc. with mutual understanding and respect, and of inclusive attitudes.

Program Specific Objectives (PSO)

1. Identify, understand, formulate and analyze complex engineering problems in the field of Data Analysis, Big Data, Database Management, Predictive Analysis, Trends Identification and Identifying Business Insights.
2. Acquire, Store, Retrieve, Process and finally convert data into knowledge in the field of artificial intelligence, data mining, network management and security, and Internet of Things applications through use of secure, reliable and cost effective state of art Analysis tools efficiently.

Lab Objectives

Students will try to:

1. To acquaint yourself with the process of identifying the needs and converting it into the problem.

2. To familiarize the process of solving the problem in a group.

3. To acquaint yourself with the process of applying basic engineering fundamentals to attempt solutions to the problems.

4. To inculcate the process of self-learning and research.

Lab Outcomes

Student will be able to:

1. Identify problems based on societal /research needs.

2. Apply Knowledge and skill to solve societal problems in a group.

3. Develop interpersonal skills to work as a member of a group or leader.

4. Draw the proper inferences from available results through theoretical/ experimental/simulations.

5. Analyze the impact of solutions in societal and environmental context for sustainable development.

6. Use standard norms of engineering practices

7. Excel in written and oral communication.

8. Demonstrate capabilities of self-learning in a group, which leads to lifelong learning.

9. Demonstrate project management principles during project work.

Acknowledgement

A project is something that could not have been materialized without cooperation of many people. This project shall be incomplete if I do not convey my heartfelt gratitude to those people from whom I have got considerable support and encouragement.

It is a matter of great pleasure for us to have a respected **Prof. Amita Raman,** as my project guide. We are thankful to him/her for being a constant source of inspiration.

We would also like to give our sincere thanks to **Prof. Anuja Chandane, H.O.D, CSE-AIML** Department**, Prof. Amita Raman, Project co-coordinator of TE** for their kind support.

We would like to express our deepest gratitude to **Dr. Manjusha Deshmukh,** our principal of Saraswati college of Engineering, Kharghar, Navi Mumbai.

Last but not the least I would also like to thank all the staff members of Saraswati college of Engineering (CSE AIML Department) for their valuable guidance with their interest and valuable suggestions brightened us.

1) **SHRUTI PEDNEKAR (55)**
2) **CHITRANG SAWANT (62)**
3) **LABH SAWANT (63)**
4) **SNEHA SHINDE (66)**

1.INTRODUCTION

In an era of personalization, individuals increasingly seek customized solutions in all aspects of life, including fashion and style. However, determining which styles are most flattering can be a subjective and challenging process. The fundamental principle of styling often revolves around complementing an individual's natural features, with face shape being one of the most critical factors. The "FaceFit" project is born from the goal of automating and objectifying this process.

The primary objective of this project is to design, implement, and evaluate an AI-driven application that can analyze a person's face and provide scientifically grounded style advice. The system aims to bridge the gap between complex facial analysis and practical, everyday fashion choices. By employing a pipeline of advanced AI techniques, the application demystifies styling rules and empowers users with personalized knowledge.

This report details the complete lifecycle of the FaceFit project, from data acquisition and model training to system architecture and final deployment. It covers the core technologies utilized, including OpenCV for image processing, Dlib for facial landmark detection, Scikit-learn for machine learning model development, and the Google Gemini API for generating rich, descriptive content.

2.LITERATURE SURVEY

Virtual Try-On System Using MediaPipe and OpenCV for AI (2025) by N Krishnapriya: This paper proposes a solution for virtual try-on specifically focusing on improving the blend and fit accuracy of virtual accessories in Augmented Reality environments. The core methodology utilizes the MediaPipe framework for robust human pose estimation and OpenCV for real-time image processing. MediaPipe is valuable for establishing stable key points on the user's body, which is a necessary precursor for accurately placing virtual items.

VIRTUAL TRY ON (2025) by Vaishnavi P, et al.: This work presents a web-based try-on system that integrates deep learning techniques with traditional image processing for blending clothing images onto users. A key focus is on effective segmentation and background removal, essential for clean overlaying of virtual apparel. The methods outlined here are relevant to the virtual try-on component of the Face-Fit AI system, particularly for the Accessory Overlay Path in our algorithm.

Image-Based Virtual Try-On: A Survey (2023) by Dan Song, et al.: This paper offers a systematic review of image-based virtual try-on solutions, covering entire pipelines, methods for human representation, and clothing warping techniques. It is critical for understanding the challenges in creating realistic virtual fittings and provides unified evaluation metrics and open-source resources. The survey highlights the complexity of accessory warping, which directly informed the decision to use Dlib's robust landmark detection for head pose estimation in the Face-Fit AI system.

A Comprehensive Survey on Deep Facial Expression Recognition (2023) by M Sajjad: Although focused on expression recognition (FER), this survey is crucial for understanding advanced facial analysis applications in AR environments. It provides an overview of trends, challenges, and key datasets used in deep learning for facial tasks. The techniques for facial landmark detection and feature processing outlined in FER research provided a strong foundation for the feature engineering pipeline used in the Face-Fit AI classifier.

Novelty and Contribution of Face-Fit AI

Existing literature heavily focuses on pure computer vision for virtual try-on. However, the Face-Fit AI system introduces a **hybrid methodology** that combines established geometric feature classification (using Dlib and Random Forest) with the advanced reasoning capabilities of a **Generative AI model (Google Gemini API)**. This integration moves the project beyond a simple virtual fitting utility to a **personalized style advisor**, generating rich, descriptive, and nuanced recommendations that traditional ML classification cannot provide.

Title	Year of Publication	Author(s)	Description
Virtual Try-On System Using MediaPipe and OpenCV for AI	2025	N Krishnapriya	Proposes a solution for virtual try-on utilizing MediaPipe for pose estimation and OpenCV for real-time image processing, improving blend and fit accuracy in AR environments.
VIRTUAL TRY ON	2025	Vaishnavi P, Kalaiyarasi K, Shivani S, Subashini E, Dr. A. Punitha	Presents a web-based try-on system using deep learning and traditional image processing to blend clothing images onto users, highlighting segmentation and background removal techniques
Image-Based Virtual Try-On: A Survey	2023	Dan Song, Xuanpu Zhang, Juan Zhou, Weizhi Nie, Ruofeng Tong, Mohan Kankanhalli	Systematic review covering pipelines, human representation, clothing warping, and challenges for image-based virtual try-on solutions, with unified evaluation metrics and open-source resources
A Comprehensive Survey on Deep Facial Expression Recognition	2023	M Sajjad	Surveys FER techniques used in facial AR applications, outlining trends, challenges, and key datasets for recognition tasks

3.PROBLEM STATEMENT

The central challenge addressed by this system is the subjectivity, friction, and lack of personalization inherent in the traditional process of accessory selection.

Subjective Selection: Consumers often rely on trial-and-error, store mirrors, or inconsistent opinions from friends when choosing accessories. This subjective process frequently results in dissatisfaction, leading to suboptimal purchases and increasing the cognitive burden on the consumer.

Time and Hygiene Constraints: In physical retail environments, physically trying on multiple items like glasses or hats is time-consuming. Furthermore, it introduces minor hygiene concerns related to product handling by multiple customers.

Commercial Inefficiency: For e-commerce platforms, the inability to virtually fit products leads directly to high return rates (often exceeding 25% for apparel accessories), significantly impacting logistical costs and profitability.

Need for Objectivity: There is a substantial need for an objective, scientific, and rapid method to analyze individual facial structures and determine the styles (shape, size, contour) that are best-suited to create a balanced and flattering appearance.

The primary objective of this project is to develop a reliable, low-latency, and accessible tool that provides instant face shape analysis and accurate virtual fitting, thereby dramatically improving the user’s accessory selection process, boosting consumer confidence, and offering a scalable model for retail technology integration.

4.PROPOSED SYSTEM

The architecture of the Face Fit system is designed as a modular pipeline, ensuring a logical flow of data from initial input to the final recommendation output. The system comprises four primary modules: the User Interface (UI), Computer Vision, Machine Learning (ML) Classification, and the Recommendation Engine.

4.1 COMPUTER VISION AND FEATURE ENGINEERING

The Computer Vision Module is the core of the facial analysis pipeline, responsible for translating a user's image into quantifiable data for the classification model.

4.1.1 Landmark Detection and Feature Extraction

Upon receiving an image, the system utilizes Dlib's frontal face detector and the pre-trained shape_predictor_68_face_landmarks.dat model to accurately map 68 key points (landmarks) on the face. These raw coordinates are then processed by the face_feature_extractor.py script to generate a normalized feature vector, which is essential for accurate face shape classification.

4.1.2 Geometric Feature Calculation and Normalization

The critical step in Feature Engineering is converting the raw pixel locations of the 68 landmarks into a scale-invariant feature vector. This involves calculating a series of normalized metrics, including:

- Facial Ratios: These include the ratio of face length to face width, and the ratio of jaw width to cheekbone width. These metrics are fundamental in differentiating between major face shapes (e.g., Oval vs. Round vs. Heart).
- Key Angles: Specific angles, such as the Jaw Angle and Chin Angle, are calculated. The Jaw Angle is crucial for differentiating between 'Round' and 'Square' shapes, while the Chin Angle helps in identifying 'Heart' shapes.

Normalization Technique: To ensure that the feature set is invariant to the scale of the image or the subject's distance from the camera, all distances and ratios are normalized using the inter-ocular distance (the distance between the eyes). This technique ensures consistency in classification regardless of image input quality.

4.2 MACHINE LEARNING CLASSIFICATION

The normalized feature vector is passed to the ML classifier, which uses the face_shape_classifier.pkl file, a trained Random Forest Classifier. A Random Forest model was chosen for its proven robustness, ability to handle complex, non-linear relationships in geometric data, and high classification accuracy. This model predicts the user's face shape (e.g., Oval, Round, Square) and gender.

4.3 GENERATIVE AI RECOMMENDATION ENGINE

The final style advice is generated by a Hybrid Recommendation Engine, which combines local lookup with Generative AI for nuanced personalization.

4.3.1 Google Gemini API Integration

For generating detailed and personalized reports, the system leverages the power of the Google Gemini Large Language Model (LLM) via the google-generativeai API.

- Structured Prompting: The classified face shape and gender are used to construct a precise, detailed prompt that guides the AI's response style. The prompt instructs the model to act as a "world-class personal stylist" and requires it to address four specific areas:

    1. Overall vibe and key facial feature commentary.
    2. Suggestion of two distinct eyewear styles with justification.
    3. Recommendation of two flattering hairstyles.
    4. Recommendation of a palette of 4-5 complementary colors.

- Caching and Fallback Mechanism: To optimize performance and reduce operational costs, responses from the Gemini API are cached in the gemini_cache.json file. The application prioritizes using a cached result based on the shape and gender before making a new API call. If the API key is missing or the call fails, a local fallback report is displayed, ensuring a reliable user experience.

This expansion provides the depth required for a Project Report, specifically detailing the algorithms, the reasoning behind the ML model choice, and the innovative integration of Generative AI.

4.1 ALGORITHM

Algorithm for Real-Time Face Analysis and Augmentation

This algorithm describes the continuous process for detecting a face, classifying its shape, and applying an augmented reality accessory overlay.

Step 1: START / INITIALIZE MODELS

Action: Begin the application and load necessary resources.

Description: Initialize the face detector, 68-landmark predictor, and the trained Machine Learning (ML) model for face shape classification.

Step 2: INPUT FRAME

Action: Capture the next frame of data.

Description: Input a new image frame from the webcam or a static image file.

Step 3: DECISION: DETECT FACE?

Action: Run the face detection model.

Description: Check if one or more human faces are present in the current frame. If YES, proceed to Step 4. If NO, skip to Step 10.

Step 4: FIND 68 FACIAL LANDMARKS

Action: Locate 68 distinct points on the detected face.

Description: Use the landmark predictor model to map the key geometric points of the face.

Next Step: Simultaneously proceed to Step 5 (Classification Path) and Step 7 (Augmentation Path).

Classification Path (Face Shape)

Step 5: EXTRACT GEOMETRIC FEATURES (RATIOS)

Action: Calculate facial measurements.

Description: Use the 68 landmarks to compute specific distance ratios (e.g., width-to-length ratios) that define the face shape.

Step 6: ML MODEL: CLASSIFY FACE SHAPE

Action: Run the classification model.

Description: Input the geometric features into the trained ML model to output the face shape result (e.g., 'Oval', 'Square').

Step 7: DISPLAY FACE SHAPE & SUGGESTIONS

Action: Prepare face shape output for the user.

Description: Generate text/graphics displaying the classified face shape and tailored style suggestions.

Next Step (Merge): Go to Step 12.

Augmentation Path (Accessory Overlay)

Step 8: CALCULATE HEAD POSE ANGLE & SCALE

Action: Determine 3D orientation and size.

Description: Use the landmarks to calculate the head's rotation (pitch, yaw, roll) and scale relative to the camera.

Step 9: APPLY ROTATION & SCALING TO ACCESSORY IMAGE

Action: Transform the accessory graphic.

Description: Adjust the virtual accessory (e.g., sunglasses, hat) to match the calculated head pose and scale.

Step 10: OVERLAY ACCESSORY ONTO FACE USING ALPHA BLENDING

Action: Merge the accessory and the frame.

Description: Combine the transformed accessory with the original input frame, using alpha blending for realistic transparency and integration.

Next Step (Merge): Go to Step 12.

Error Handling and Loop

Step 11: NO FACE DETECTED

Action: Skip processing.

Description: No landmarks or calculations are performed. The original frame (or a 'No Face' message) is maintained.

Next Step (Merge): Go to Step 12.

Step 12: DISPLAY AUGMENTED FRAME TO USER

Action: Present the final output.

Description: Display the processed image to the user. This frame includes the augmented accessory overlay (if applicable) and the face shape information/suggestions.

Step 13: DECISION: USER QUITS LOOP?

Action: Check for user termination.

Description: Continuously check for a user input (e.g., clicking a 'Stop' button). If NO (Continue), go back to Step 2. If YES (Stop), go to Step 14.

Step 14: STOP APPLICATION

Action: Gracefully shut down.

Description: De-initialize the webcam feed and models, then terminate the application.

4.2 FLOWCHART

The application starts by initializing the models. It then enters a loop, taking a Webcam Frame or Static Image as input.

Face Detection and Analysis

1. Detect Face: The system first checks for a face in the image.
   - No: If no face is detected, it returns a "No Face Detected" message and continues the loop for the next input.
   - Yes: If a face is found, it proceeds to "Find 68 Facial Landmarks".
2. Facial Data Processing: The 68 landmarks are used for two parallel processes:
   - Face Shape Classification: Geometric Features (Ratios) are extracted, which are then fed into an ML Model to Classify Face Shape (e.g., Oval). This result is then used to Display Face Shape & Suggestions.
   - Accessory Augmentation: The landmarks are used to Calculate Head Pose Angle & Scale. This data is used to Apply Rotation & Scaling to Accessory Image and then Overlay Accessory onto Face using Alpha Blending.

Output and Termination

1. Display: The results from both parallel paths converge to Display Augmented Frame to User, showing the accessory overlay and the face shape information/suggestions.
2. Loop Control: After displaying the result, the application checks if the "User Quits Loop?"
   - No (Continue): It returns to the start of the loop to process the next frame/image.
   - Yes (Stop): The application proceeds to Stop Application.

In essence, the flowchart describes an iterative system that detects a face, analyzes its shape, and simultaneously overlays a virtual accessory, before displaying the combined, augmented result to the user.

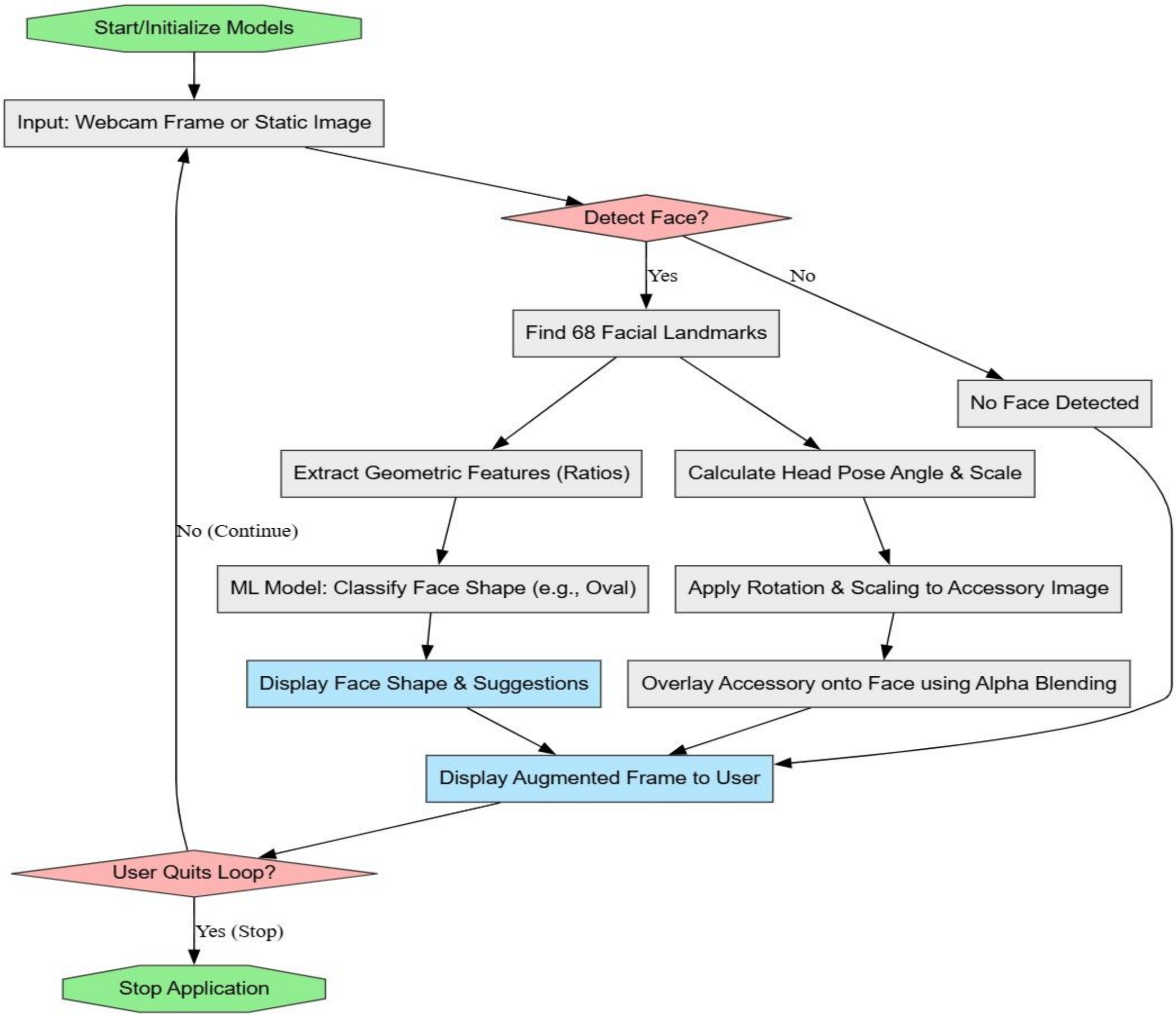

Fig 4.2.1 Flowchart of FcaeFIt

This flowchart details the steps of an application that initializes machine learning models and then enters a loop to process a webcam frame or static image.

5 CODE

```
# --- RELIABLE GEOMETRIC FACE SHAPE ENGINE ---
import math


def get_face_shape_geometric(landmarks, frame_w, frame_h):
    """Calculates face shape based on landmark coordinates."""


    # Key landmark points for calculation
    p_jaw_l, p_jaw_r = landmarks[172], landmarks[397]
    p_cheek_l, p_cheek_r = landmarks[234], landmarks[454]
    p_forehead, p_chin = landmarks[10], landmarks[152]


    # Calculate critical distances
    face_width_jaw = math.hypot((p_jaw_l.x - p_jaw_r.x) * frame_w, (p_jaw_l.y -
p_jaw_r.y) * frame_h)
    face_width_cheeks = math.hypot((p_cheek_l.x - p_cheek_r.x) * frame_w,
(p_cheek_l.y - p_cheek_r.y) * frame_h)
    face_height = math.hypot((p_forehead.x - p_chin.x) * frame_w, (p_forehead.y -
p_chin.y) * frame_h)


    # Logic for classifying face shape
    face_shape = "Oval" # Default shape
    if face_width_jaw > 0 and face_height > 0:
```

```
        if abs(face_width_cheeks - face_height) < 25:
            face_shape = "Round"
        elif abs(face_width_jaw - face_width_cheeks) < 30:
            face_shape = "Square"
        elif face_width_cheeks > face_width_jaw and (face_height / face_width_cheeks) < 1.3:
            face_shape = "Heart"
    return face_shape
# --- GEMINI VISION REPORTING ENGINE ---
import google.generativeai as genai


def get_gemini_vision_report(api_key, image, face_shape, gender):
    """Generates a style report by sending the user's image and data to the Gemini API."""
    try:
        genai.configure(api_key=api_key)
        model = genai.GenerativeModel('models/gemini-2.5-pro')


        # The detailed prompt guides the AI's response style and content.
        prompt = f"""Act as a world-class personal stylist writing an exclusive style note for a client.
        Your tone is chic, insightful, and warmly personal...
```

```
        I've determined their face shape is **{face_shape}** and they identify as
**{gender}**.

        Based on the image, write a style guide for them.

        1. Comment on their overall vibe and a key feature.

        2. Suggest two distinct eyewear styles and explain why they work.

        3. Recommend two flattering hairstyles.

        4. Recommend a palette of 4-5 complementary colors based on their features.

        Conclude with a warm sign-off."""


        response = model.generate_content([prompt, image])

        return response.text

    except Exception as e:

        # Returns an error message if the API call fails

        return f"An error occurred. Please check your internet connection or API key.
Error: {e}"

# --- Caching and Fallback Snippet from _get_and_display_report ---


def _get_and_display_report(self, img, shape, gender):

    """Orchestrates the report generation with caching and fallback."""

    cache_key = f"{shape.lower()}_{gender.lower()}"

    cache_data = self._load_cache()


    if cache_key in cache_data:
```

```
        # 1. Use cached result if available
        report = f"--- Displaying Cached Style Profile ---\n\n{cache_data[cache_key]}"
        self.rep_txt.insert(tk.END, report)

    elif not GEMINI_API_KEY:
        # 2. Use local fallback if API key is missing
        fallback_report = self.get_local_style_report(shape, gender)
        report = f"NOTE: AI Advisor offline.\n\n--- Displaying Local Suggestions ---
\n\n{fallback_report}"
        self.rep_txt.insert(tk.END, report)

    else:
        # 3. Call the Gemini API as a last resort
        try:
            api_report = get_gemini_vision_report(GEMINI_API_KEY, img, shape,
gender)
            if "An error occurred" in api_report: raise Exception(api_report)

            # Save the new result to the cache
            cache_data[cache_key] = api_report
            self._save_cache(cache_data)
            self.rep_txt.insert(tk.END, api_report)
```

```
        except Exception as e:
            # 4. Use local fallback if API call fails
            fallback_report = self.get_local_style_report(shape, gender)
            report = f"NOTE: AI Advisor unavailable.\n(Error: {e})\n\n--- Displaying Local Suggestions ---\n\n{fallback_report}"
            self.rep_txt.insert(tk.END, report)

# --- Main Application Structure ---
class FaceFitApp(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("FaceFit - AI Advisor (Gemini Edition)")

        # --- State Variables ---
        self.video_stream = None
        self.last_frame_raw = None      # Stores the original camera frame
        self.last_landmarks = None      # Stores the last detected face landmarks
        self.selected_accessories = {}  # Holds data for virtual try-on items

        # --- Build UI ---
        self._build_ui() # This method sets up all the Tkinter widgets

    # --- Core Methods ---
    def start_webcam(self):
```

```
        """Initializes and starts the webcam capture thread."""
        pass # Implementation details omitted for brevity

    def process_frame_for_tryon(self, frame):
        """Detects landmarks and overlays selected accessories on a given frame."""
        pass # Implementation details omitted for brevity

    def run_gemini_analysis(self):
        """The main function triggered by the user to start the AI analysis."""
        # 1. Get the current image (from webcam or file).
        # 2. Detect face landmarks.
        # 3. Calculate the geometric face shape.
        # 4. Start a new thread to call _get_and_display_report to avoid freezing the UI.
        pass # Implementation details omitted for brevity

    def _on_close(self):
        """Ensures the webcam thread is stopped cleanly when closing the app."""
        self.stop_webcam()
        self.destroy()
if __name__ == "__main__":
    app = FaceFitApp()
    app.mainloop()
```

6.PROJECT OUTPUT

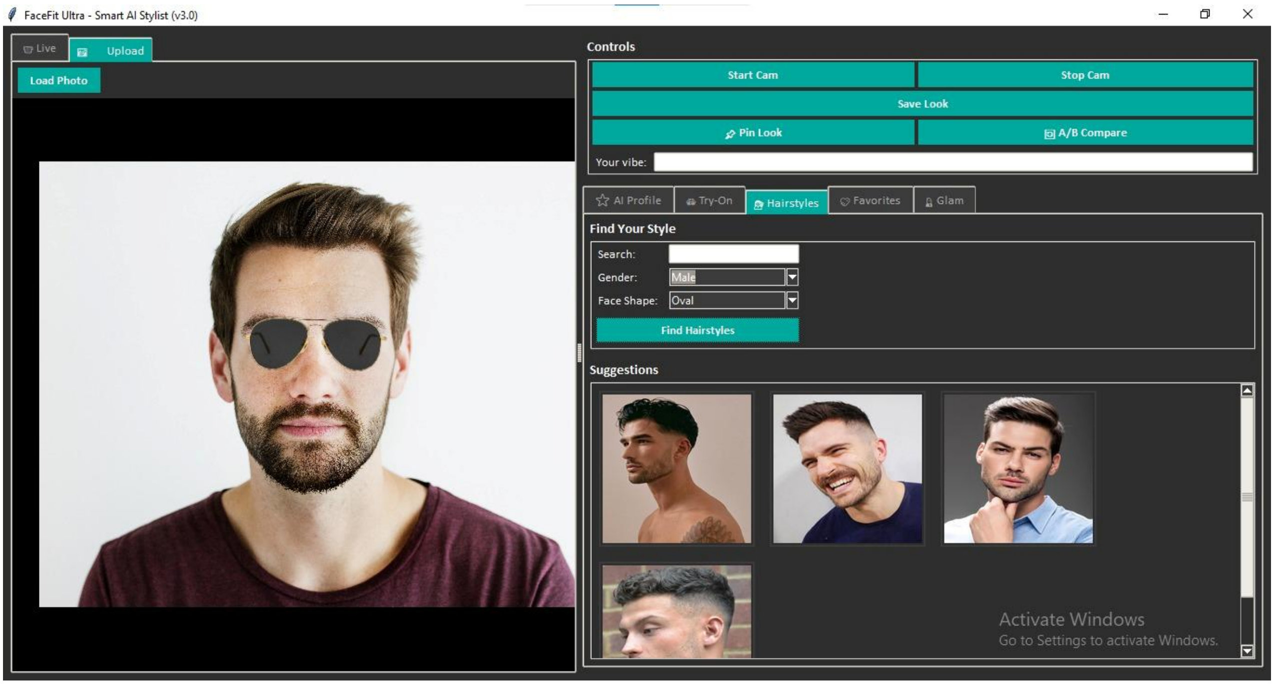

Fig 6.1 This image is a screenshot of the FaceFit Ultra AI Stylist software's "Hairstyles" recommendation feature.

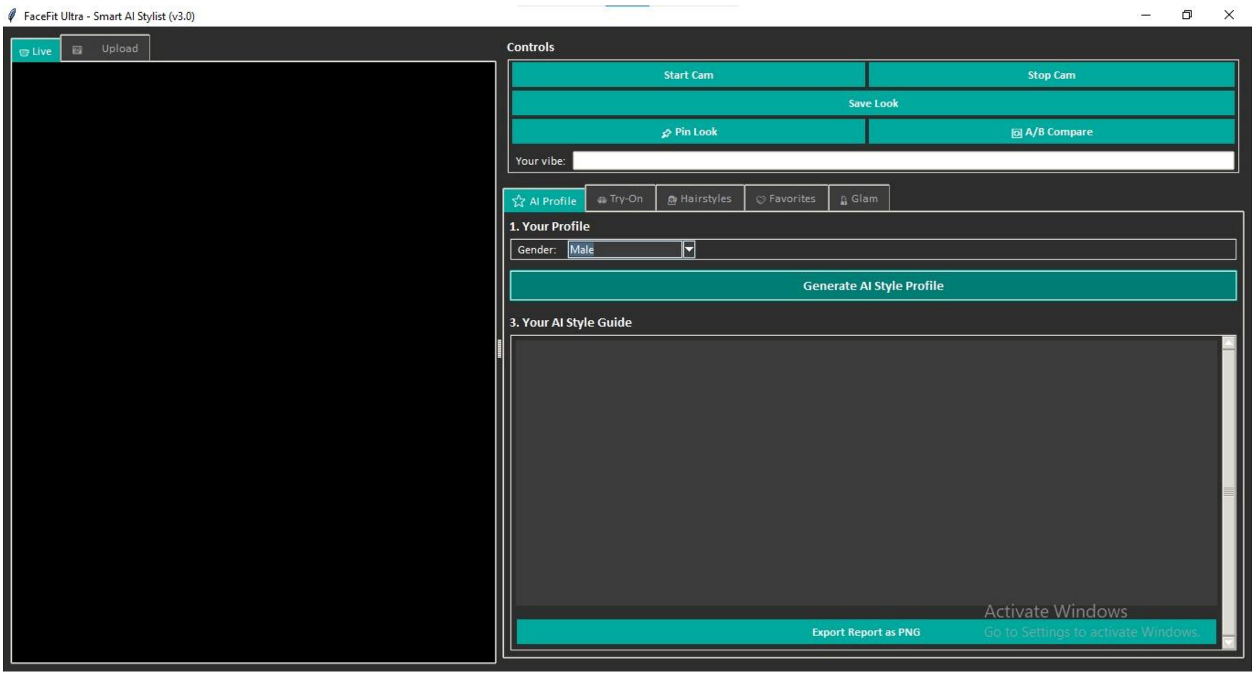

Fig 6.2 This image is a screenshot of the FaceFit Ultra AI Stylist software's "AI Profile" homepage in an initial state.

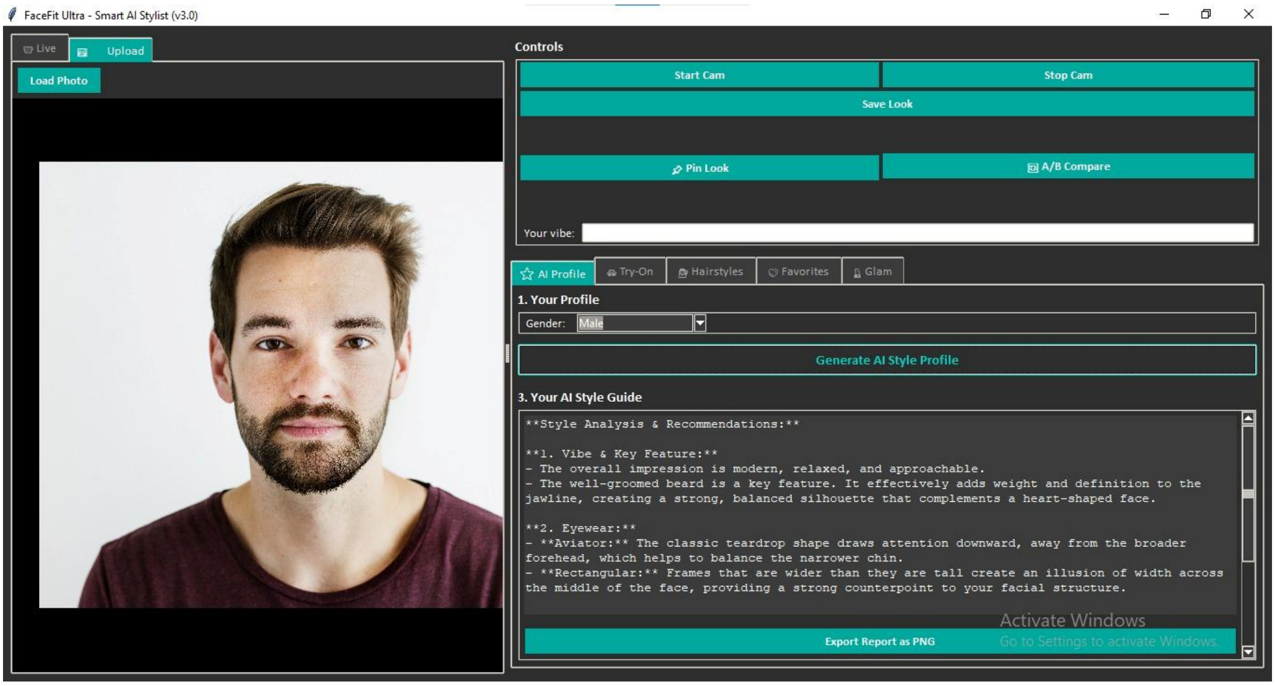

Fig 6.3This image displays the output of the FaceFit Ultra AI Stylist software, showing a male model's photo alongside its AI Style Guide.

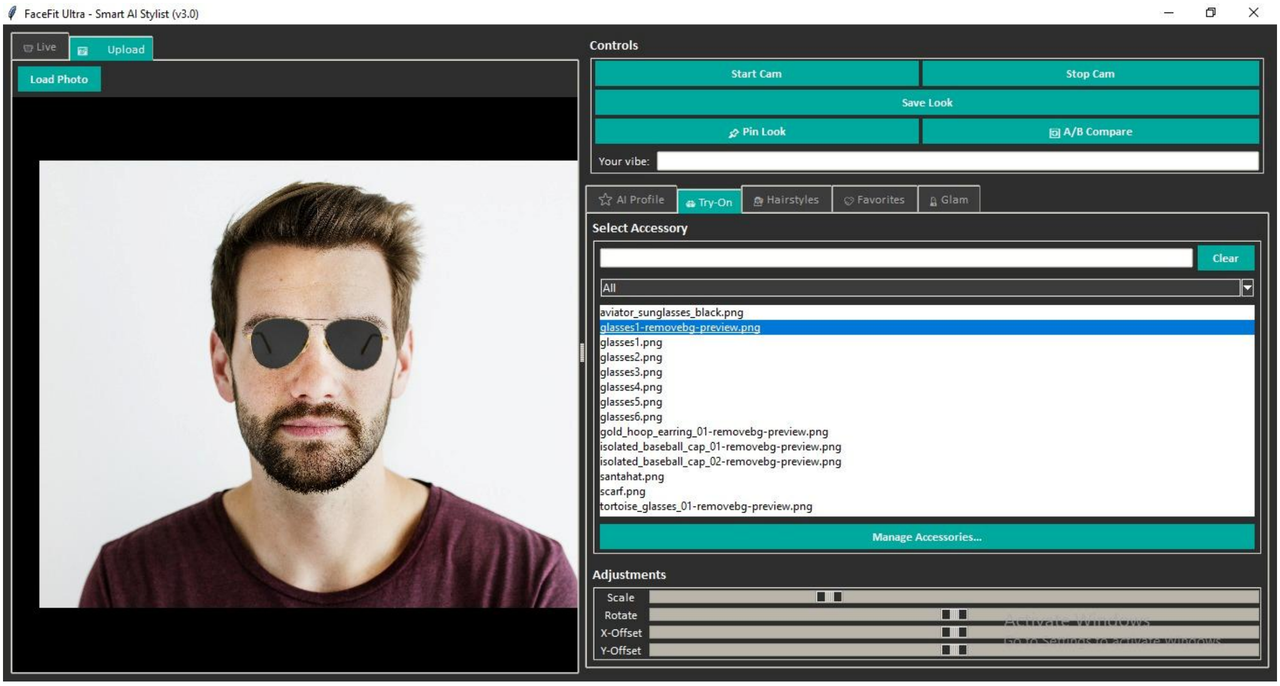

Fig 6.4 This image is a screenshot of the FaceFit Ultra AI Stylist software's "Try-On" feature

7.CONCLUSION

The FaceFit project successfully achieved its objective of creating an intelligent personal stylist. By combining computer vision and machine learning, it provides a data-driven approach to style recommendations, moving beyond generic advice to offer personalized guidance.

There are several avenues for future enhancement:

- Expanded Feature Set: Incorporating analysis of other features like skin tone, eye color, and hair type to provide even more comprehensive recommendations (e.g., makeup, clothing colors).
- Advanced Recommendation Models: Implementing a more sophisticated recommendation engine, perhaps using collaborative filtering or deep learning models trained on fashion datasets.
- Real-time Video Analysis: Modifying the application to work with a live video feed, providing real-time feedback and virtual try-on features for different hairstyles or glasses.
- Mobile Application: Developing a mobile version of FaceFit to make the tool more accessible to a wider audience.

8.REFERENCES

[1] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Computer Vision and Pattern Recognition, 2001. CVPR 2001.

[2] Kazemi, V., & Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (The basis for Dlib's shape predictor).

[3] Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools.

[4] Standard Face Shape Classification Methods and Geometric Feature Extraction Techniques in Biometrics.

[5] Python Libraries: Dlib Documentation, OpenCV Documentation, Scikit-learn Documentation, Tkinter Reference.

[6] https://www.youtube.com/watch?v=swSp8ksR2Q

[7] https://www.youtube.com/watch?v=VjDsJ2piXg8

[8] https://www.youtube.com/watch?v=azvA2Bn2aXw

[9] https://www.youtube.com/watch?v=C94pTaKoLbU