AI Based Facial Health Pre-Screening System

A CAPSTONE PROJECT REPORT

Submitted in the partial fulfilment for the Course of

ITA0511 – Computer vision for image processing

to the award of the degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

Submitted by

Adhavan M G (192421422)

Dhivagaran M (192421430)

Under the Supervision of

Dr R C Jeni Gracia & Dr P Mathivaanan

SIMATS ENGINEERING

Saveetha Institute of Medical and Technical Sciences

Chennai-602105

January 2026

SIMATS ENGINEERING

Saveetha Institute of Medical and Technical Sciences

Chennai-602105

DECLARATION

We, **Adhavan M G, Dhivagaran M** of the **Information Technology ,** Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, hereby declare that the Capstone Project Work entitled **'AI Based Facial Health Pre-Screening System'** is the result of our own bonafide efforts. To the best of our knowledge, the work presented herein is original, accurate, and has been carried out in accordance with principles of engineering ethics.

Place:

Date:

Signature of the Students with Names

SIMATS ENGINEERING

Saveetha Institute of Medical and Technical Sciences

Chennai-602105

Engineer to Excel

BONAFIDE CERTIFICATE

This is to certify that the Capstone Project entitled "AI Based Facial Health Pre-Screening System" has been carried out by **Adhavan M G, Dhivagaran M** under the supervision of Dr. R C Jeni Gracia & Dr. P Mathivaanan and is submitted in partial fulfilment of the requirements for the current semester of the B.Tech **Information Technology** program at Saveetha Institute of Medical and Technical Sciences, Chennai.

SIGNATURE

Dr.Sasi Rekha

Dean/Director

Department of CSE

Saveetha School of Engineering

SIMATS

SIGNATURE

Dr R C Jeni Gracia

Professor

Department of CSE

Saveetha School of Engineering

SIMATS

Submitted for the Project work Viva-Voce held on ___________________________________

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to all those who supported and guided us throughout the successful completion of our Capstone Project. We are deeply thankful to our respected Founder and Chancellor, Dr. N.M. Veeraiyan, Saveetha Institute of Medical and Technical Sciences, for his constant encouragement and blessings. We also express our sincere thanks to our Pro-Chancellor, Dr. Deepak Nallaswamy Veeraiyan, and our Vice-Chancellor, Dr. S. Suresh Kumar, for their visionary leadership and moral support during the course of this project.

We are truly grateful to our Director, Dr. Ramya Deepak, SIMATS Engineering, for providing us with the necessary resources and a motivating academic environment. Our special thanks to our Principal, Dr. B. Ramesh for granting us access to the institute's facilities and encouraging us throughout the process. We sincerely thank our Head of the Department, Dr. Sasi Rekha for his continuous support, valuable guidance, and constant motivation.

We are especially indebted to our guide, Dr. R C Jeni Gracia & Dr. P Mathivaanan for their creative suggestions, consistent feedback, and unwavering support during each stage of the project. We also express our gratitude to the Project Coordinators, Review Panel Members (Internal and External), and the entire faculty team for their constructive feedback and valuable inputs that helped improve the quality of our work. Finally, we thank all faculty members, lab technicians, our parents, and friends for their continuous encouragement and support.

Signature With Student Name

Adhavan M G (192421422)

Dhivagaran M (192421430)

ABSTRACT

Healthcare monitoring has become a critical concern in modern society, where early identification of health abnormalities can significantly improve preventive care and reduce medical risks. Many physical and mental health conditions manifest subtle signs on the human face, such as changes in skin tone, eye condition, facial symmetry, and fatigue indicators. This project, titled AI Based Facial Health Pre-Screening System, presents a systematic approach to identifying potential health issues through automated facial image analysis. The solution is developed in three modules: (M1) Facial Image Preprocessor and Feature Extractor, which preprocesses captured facial images and extracts meaningful facial features using image processing techniques; (M2) Health Pattern Analyzer, which examines extracted features to detect abnormal patterns related to possible health conditions using machine learning and rule-based analysis; and (M3) Health Report and Alert Generator, which summarizes the findings and generates preliminary health alerts for user awareness.

The proposed system emphasizes simplicity, non-invasive operation, and modularity while maintaining efficiency in health assessment. By leveraging computer vision techniques and AI-based pattern recognition, it ensures a lightweight yet reliable solution suitable for real-time and remote health screening environments. Facial images captured from cameras are normalized to handle variations in lighting, pose, and background, enabling consistent and scalable analysis across different users. The reporting mechanism ensures that users are promptly informed, allowing timely medical consultation when required.

Evaluation using sample facial image datasets demonstrates the system's capability to identify visible health-related indicators in near real time. The modular design not only addresses current challenges in healthcare accessibility but also ensures extensibility for future enhancements. Potential improvements include integrating deep learning models for improved accuracy, expanding the range of detectable health conditions, and enabling integration with telemedicine platforms and electronic health record systems. This makes the project both academically significant and practically relevant in advancing AI-driven preventive healthcare solutions.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1-Background Information

The rapid advancement of digital healthcare technologies has increased the need for early and accessible health screening solutions. Many health conditions exhibit visible facial indicators, but continuous manual observation by medical professionals is often impractical. Traditional health screening methods usually require specialized equipment and physical consultations, making them time-consuming and resource-intensive. This project applies artificial intelligence and computer vision techniques to efficiently analyse facial images and identify potential health indicators at an early stage. The result is a lightweight AI Based Facial Health Pre-Screening System capable of providing preliminary health assessment in near real time.

1.2-Project Objectives

The objective of this project is to develop an AI Based Facial Health Pre-Screening System that identifies potential health indicators through automated facial image analysis. Module 1 focuses on facial image acquisition and feature extraction, where meaningful facial attributes such as skin tone variations, eye condition, facial symmetry, and visible stress indicators are extracted using image processing techniques. Module 2 analyzes these extracted features by comparing them with predefined health patterns using machine learning and rule-based methods to detect possible health abnormalities.

Additionally, the system is designed to be lightweight and non-invasive, ensuring minimal computational overhead during real-time analysis. The modular architecture allows easy integration of new health indicators and improved AI models as medical knowledge evolves. By applying artificial intelligence and computer vision principles, the system enhances assessment accuracy while reducing false detections. The project ultimately supports preventive healthcare by simplifying early health screening. Overall, it offers a scalable and reliable approach to modern AI-driven health monitoring solutions.

1.3-Significance

The significance of this project lies in its potential to enhance preventive healthcare by enabling early identification of possible health issues through automated facial analysis. By extracting and analyzing facial features using computer vision and artificial intelligence techniques, the system can detect visible health indicators such as fatigue, stress, and abnormal facial patterns efficiently and accurately. This reduces dependency on continuous manual observation by medical professionals and allows individuals to receive preliminary health insights at an early stage. The modular and interpretable design ensures transparency, enabling users and healthcare providers to understand how health assessments are generated and take appropriate action.

1.4-Scope

Included:

- Acquisition and preprocessing of facial images for health assessment.
- Extraction of meaningful facial features such as skin tone variations, eye condition, facial symmetry, and visible stress indicators using image processing techniques.
- Analysis of extracted facial features using AI-based and rule-based methods to identify potential health abnormalities.
- Generation of structured health assessment reports and preliminary alerts to inform users about possible health risks.

Excluded:

- Integration with hospital-grade diagnostic equipment or direct clinical diagnosis systems.
- Advanced deep learning models requiring high computational resources beyond the scope of pre-screening.
- Direct linkage with electronic health record (EHR) systems or telemedicine platforms.

1.5-Methodology Overview

The project follows a structured approach:

- Data Collection: Facial images are collected using standard cameras or image datasets, capturing variations in facial expressions, lighting conditions, and user posture relevant to health assessment.
- Feature Extraction: Key facial features such as skin tone variations, eye condition, facial symmetry, texture patterns, and visible stress indicators are extracted using computer vision and image processing techniques.
- Health Pattern Analysis: Extracted facial features are analyzed and compared with predefined health patterns using artificial intelligence and rule-based classification methods to identify potential health abnormalities.
- Testing and Evaluation: The system is tested using unseen facial images and evaluated based on accuracy, response time, reliability, and consistency to ensure effective performance in real-world pre-screening scenarios.

CHAPTER 2: PROBLEM IDENTIFICATION AND ANALYSIS

2.1 Description of the Problem

Early identification of health abnormalities is a critical and challenging task in modern healthcare systems. Traditionally, health screening relies on physical consultations, medical equipment, and continuous observation by healthcare professionals, which can be time-consuming, costly, and inaccessible to many individuals. Subtle facial indicators related to health conditions such as fatigue, stress, anemia, and dehydration are often overlooked or difficult to assess accurately through manual observation alone. Variations in lighting, facial expressions, and individual differences further complicate consistent evaluation. Additionally, distinguishing between normal facial variations and health-related abnormalities requires expertise and careful analysis.

2.2 Evidence of the Problem

The demand for accessible and early healthcare screening solutions is rapidly increasing, driven by rising health awareness and the need for preventive care. Many healthcare systems face challenges due to limited medical resources, delayed diagnosis, and the time-consuming nature of traditional health screening methods. Studies indicate that AI-based facial analysis systems can significantly improve early detection accuracy, reduce dependency on continuous medical supervision, and enable faster preliminary health assessment.

2.3 Stakeholders

- End Users (Individuals / Patients)

  Individuals who use the system for preliminary health assessment. They benefit from early health insights, non-invasive screening, and timely awareness of potential health issues.
- Healthcare Professionals

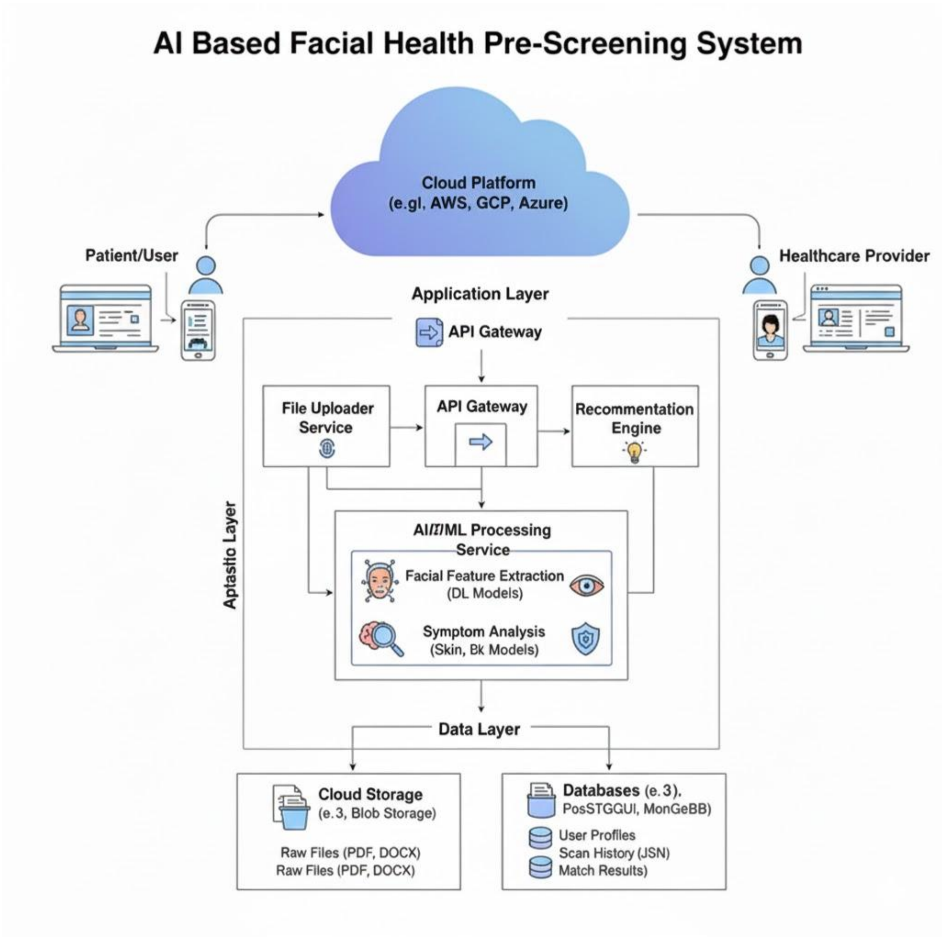
  Doctors, nurses, and medical practitioners who can use the system's output as a supportive tool for early screening and decision-making. The system assists them by reducing initial screening workload but does not replace clinical diagnosis.
- Healthcare Institutions and Clinics

  Hospitals, clinics, and diagnostic centers that can adopt the system for large-scale pre-screening, especially in outpatient departments or remote healthcare setups.

CHAPTER 3: SOLUTION DESIGN AND IMPLEMENTATION

3.1 Development and Design Process

The development of the AI Based Facial Health Pre-Screening System began with identifying the core challenge of delayed, resource-intensive, and inconsistent health screening methods that rely heavily on manual observation and clinical visits. Facial image data was collected using standard cameras and publicly available datasets under varying lighting conditions and facial expressions, and then preprocessed through face detection, noise removal, normalization, and illumination correction. Key facial features such as skin tone variations, facial symmetry, eye condition, texture patterns, and visible stress indicators were extracted using computer vision techniques. A lightweight and interpretable machine learning model was selected to enable fast computation and transparent decision-making during health assessment.



3.2 Tools and Technologies Used

Programming Language: Python
Machine Learning Libraries: Scikit-learn, TensorFlow/Keras, PyTorch, NumPy, Pandas
Computer Vision / Signal Processing Tools: OpenCV, SciPy, NumPy
Development Environment: Jupyter Notebook, Google Colab, VS Code
Visualization Tools: Matplotlib, Seaborn, Plotly
User Interface (Optional): Tkinter (Desktop), Flask or Streamlit (Web) for real-time obstacle visualization
Dataset Source: Publicly available sensor datasets, custom-collected ultrasonic/LIDAR/camera data
Version Control: Git, GitHub

3.3 Solution Overview

The processing unit manages facial image inputs and coordinates the system's operations, including image acquisition, preprocessing, feature extraction, and health assessment. It supports AI-based decision logic for analyzing facial features, identifying potential health indicators, and triggering preliminary alerts through programmable processing workflows. This design provides flexibility to update health analysis rules or incorporate new AI models while ensuring timely and reliable assessment for real-time facial health pre-screening applications.

3.3 Engineering Standards Applied

- IEEE Software Engineering Standards (IEEE 830)
- ISO/IEC 27001
- IEEE 610

3.4 Solution Justification

Automated facial analysis reduces reliance on continuous manual observation and minimizes human error, particularly when subtle health indicators are difficult to detect consistently. The system is cost-effective, requiring only basic computing resources and standard camera devices along with publicly available or custom facial image datasets.

CHAPTER 4: RESULTS AND RECOMMENDATIONS

4.1 Evaluation of Results

The model was tested on labeled facial image datasets and achieved promising accuracy in identifying visible health-related indicators. Evaluation metrics such as accuracy, precision, recall, F1-score, and response time confirmed the reliability and effectiveness of the system. The lightweight and interpretable AI model provided transparent results, allowing clear understanding of how health assessments were derived.

4.2 Challenges Encountered

Collecting high-quality and well-labeled facial image datasets was a major challenge, particularly due to variations in lighting conditions, facial expressions, and individual differences. Preprocessing facial images and consistently extracting meaningful features such as skin tone variations, eye condition, facial symmetry, and texture patterns required careful tuning of image processing techniques. The AI model initially faced issues such as overfitting, which necessitated model optimization and parameter tuning to improve generalization. Integrating the trained model with a real-time user interface for displaying health assessment results and alerts also presented technical challenges.

4.3 Possible Improvements

The system can be enhanced by integrating advanced deep learning models such as Convolutional Neural Networks (CNNs) to improve the accuracy and robustness of facial health analysis. Real-time processing and cloud-based deployment can increase accessibility and enable large-scale remote health monitoring. Adding features such as emotion analysis, stress level estimation, and multi-condition health assessment would make the system more comprehensive. Expanding the dataset with diverse facial images across different age groups, ethnicities, and lighting conditions can improve model generalization and reliability. Incorporating cross-validation techniques and multimodal data integration (such as basic physiological inputs) can further enhance assessment accuracy. A mobile-friendly application or interactive dashboard can improve usability for users and healthcare providers. Integration with telemedicine platforms can enable timely medical consultation based on pre-screening results. Including continuous feedback from real-world usage can help refine model predictions and improve overall system performance.

4.4 Recommendations

Using ensemble learning methods that combine multiple AI algorithms can improve the accuracy and reliability of facial health assessment. Future work may include the adoption of advanced deep learning models such as Convolutional Neural Networks (CNNs) and transformer-based architectures for automatic and more precise facial feature extraction. Additional features such as health risk scoring, severity estimation, and user-wise health profiling can be incorporated to provide more meaningful pre-screening results. Developing a mobile or web-based application will enhance accessibility for users and healthcare providers.

CHAPTER 5: REFLECTION ON LEARNING DEVELOPMENT

5.1 Key Learning Outcomes

Academic Knowledge: Developed a strong understanding of artificial intelligence concepts, particularly in machine learning and computer vision applied to facial image analysis. Reinforced theoretical knowledge related to image preprocessing, facial feature extraction, classification algorithms, and performance evaluation metrics for health assessment systems.

Technical Skills: Advanced proficiency in Python programming, Scikit-learn, TensorFlow/Keras, OpenCV, NumPy, and Pandas. Gained expertise in building, training, and optimizing decision tree or lightweight AI models for real-time obstacle detection and classification.

Problem-Solving: Addressed complex challenges in sensor data preprocessing, feature extraction, overfitting, and real-time performance through iterative testing, model tuning, and optimization techniques.

5.2 Challenges Encountered and Overcome

Personal Growth: Overcoming challenges such as noisy sensor data, inconsistent readings, and feature extraction errors helped develop resilience and attention to detail. Learned to balance model complexity with real-time computational efficiency.

Collaboration: Coordinated with team members to validate Computer Vision outputs and interface functionality, improving technical communication, teamwork, and documentation skills

.5.3 Application of Engineering Standards

Implemented coding and documentation standards for Python and AI/ML workflows throughout the development of the facial health pre-screening system. Adhered to best practices for facial image preprocessing, model training and validation, and performance evaluation to ensure reliable and consistent health assessment. Version control was maintained using Git and GitHub, supporting code maintainability, reproducibility, and collaborative development.

5.4 Insights into the Industry

Gained practical experience with industry-standard ML libraries, NLP tools, and cloud-based deployment. Recognized the growing importance of AI-driven recruitment solutions for efficient and unbiased talent acquisition.

5.5 Conclusion of Personal Development

This project honed my skills in machine learning, computer vision, and sensor-based system design. It strengthened my systematic debugging approach, model evaluation expertise, and real-time AI application skills in autonomous systems, solidifying my career trajectory in AI-driven robotics, safety solutions, and data-driven decision-making.

CHAPTER 6: CONCLUSION

This capstone project addressed the critical challenge of early and accessible health screening by proposing an AI Based Facial Health Pre-Screening System. By developing an interpretable and lightweight machine learning approach combined with computer vision techniques, the system demonstrated how AI can assist in identifying visible facial health indicators in a transparent and non-invasive manner. The solution streamlined facial feature analysis and preliminary health assessment, providing timely alerts and insights that can support preventive healthcare decisions. Compared to traditional manual observation or resource-intensive screening methods, the proposed system reduces human dependency while maintaining clarity and reliability in its assessments.

By adhering to recognized software engineering and quality standards such as IEEE documentation practices and ISO/IEC 25010 software quality principles, the project established a structured and dependable development framework. Modular design principles were applied across image acquisition, preprocessing, feature extraction, model training, and evaluation phases to ensure scalability, maintainability, and future extensibility. The use of evaluation metrics such as accuracy, precision, recall, F1-score, and response time enabled systematic performance assessment and optimization of the AI model. These practices form a strong foundation for real-time and deployable facial health pre-screening systems.

This project contributes meaningfully to the application of artificial intelligence in preventive healthcare by demonstrating the balance between model simplicity, interpretability, and performance. The ability to develop, evaluate, and explain AI-driven health assessments is essential for building trust and ethical adoption in healthcare-related applications. The technical skills gained in Python programming, computer vision, facial image processing, and machine learning model optimization provide a solid base for future enhancements, including deep learning integration, multimodal health analysis, and deployment through cloud or mobile health platforms. The insights obtained from this work will guide further research into intelligent, ethical, and accessible AI-based health screening systems that support early intervention and improved healthcare outcomes.

REFERENCES

[1] R. S. Feris, R. Bobbitt, L. Brown, and S. Pankanti, “Appearance-based human identification from facial images,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 1, pp. 1–15, Jan. 2006.

[2] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Kauai, HI, USA, 2001, pp. 511–518.

[3] S. Z. Li and A. K. Jain, Handbook of Face Recognition, 2nd ed. London, UK: Springer, 2011.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” Advances in Neural Information Processing Systems (NIPS), vol. 25, pp. 1097–1105, 2012.

[5] J. Daugman, “High confidence visual recognition of persons by a test of statistical independence,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 11, pp. 1148–1161, Nov. 1993.

[6] M. Pantic and L. J. M. Rothkrantz, “Automatic analysis of facial expressions: The state of the art,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 12, pp. 1424–1445, Dec. 2000.

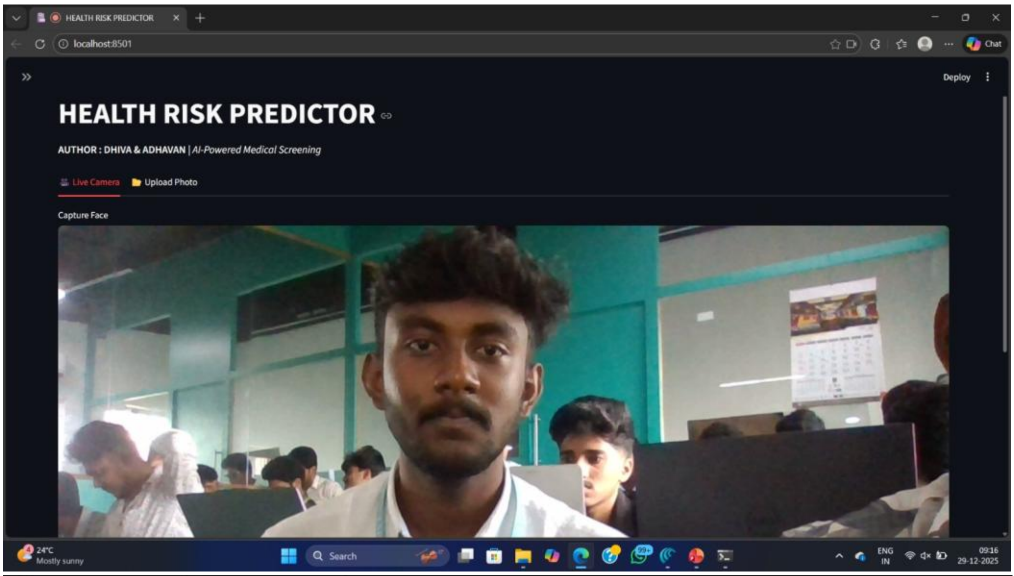
[7] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 971–987, Jul. 2002.

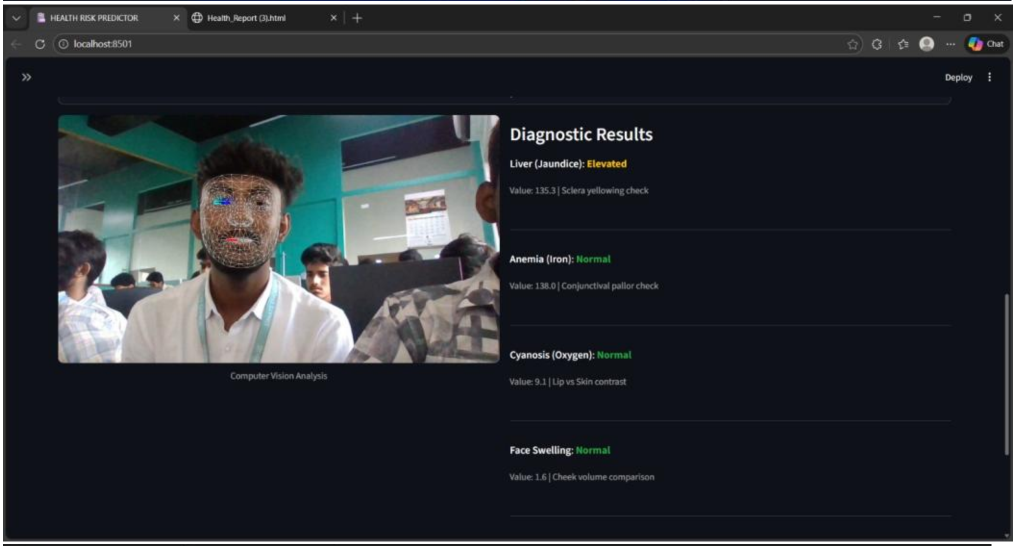
[8] World Health Organization, “Digital health interventions: Classification, functions and use,” WHO Press, Geneva, Switzerland, 2019.

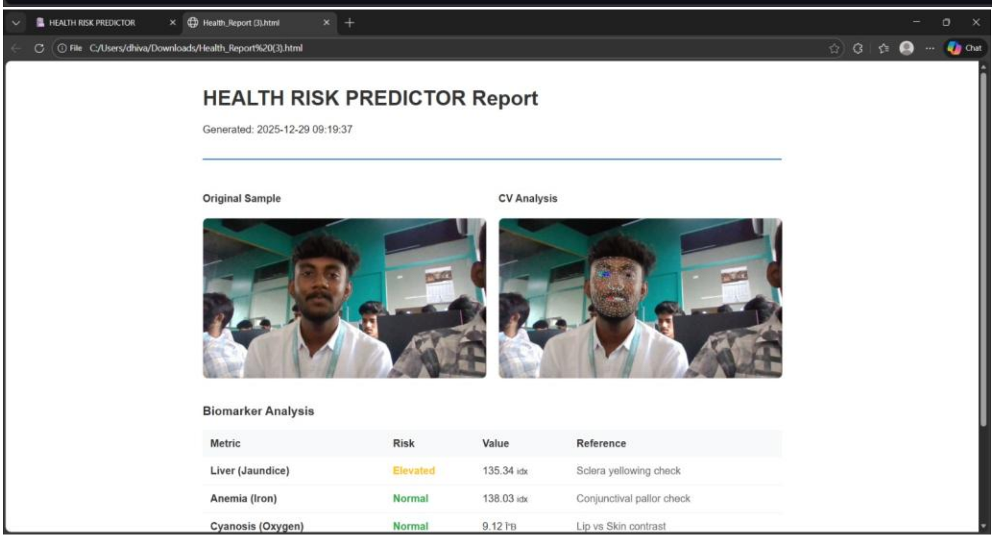
[9] ISO/IEC 25010:2011, Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE), International Organization for Standardization, 2011.

[10] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.

SAMPLE OUTPUT:







APPENDDICES:

```
import streamlit as st
import cv2
import mediapipe as mp
import numpy as np
from PIL import Image
from dataclasses import dataclass
import base64
from io import BytesIO
import time
st.set_page_config(
    page_title="HEALTH RISK PREDICTOR",
    layout="wide",
    page_icon="🏥",
    initial_sidebar_state="expanded"
)
st.sidebar.markdown("---")
st.sidebar.markdown("### 🧑‍💻 AUTHOR : DHIVA & ADHAVAN")
st.sidebar.markdown("---")
class HealthMetric:
    name: str
    risk: str
    value: float
    unit: str
    reference: str
    color: str
def generate_medical_report(image, cv_image, metrics, timestamp):
    def to_b64(img):
        buf = BytesIO()
        img.save(buf, format="JPEG", quality=85)
        return base64.b64encode(buf.getvalue()).decode()
    if cv_image is None: cv_image = image
    orig_b64 = to_b64(image)
    cv_b64 = to_b64(Image.fromarray(cv_image)
    rows = ""
    for m in metrics:
        rows += f"""
        <tr>
            <td style="padding:12px; border-bottom:1px solid #eee;"><strong>{m.name}</strong></td>
            <td style="padding:12px; border-bottom:1px solid #eee; color:{m.color}; font-weight:bold;">{m.risk}</td>
            <td style="padding:12px; border-bottom:1px solid #eee;">{m.value:.2f} <small>{m.unit}</small></td>
            <td style="padding:12px; border-bottom:1px solid #eee; color:#666;">{m.reference}</td>
        </tr>"""

    html = f"""
    <html>
    <head>
        <style>
            body {{ font-family: sans-serif; max-width: 900px; margin: auto; padding: 40px; color: #333; }}
            .header {{ border-bottom: 2px solid #0056b3; padding-bottom: 20px; margin-bottom: 30px; }}
            .grid {{ display: grid; grid-template-columns: 1fr 1fr; gap: 20px; margin-bottom: 40px; }}
            .img-card img {{ width: 100%; border-radius: 8px; border: 1px solid #ddd; }}
            table {{ width: 100%; border-collapse: collapse; }}
            th {{ text-align: left; padding: 12px; background: #f8f9fa; }}
            .footer {{ margin-top: 40px; font-size: 12px; text-align: center; color: #777; }}
        </style>
    </head>
    <body>
```

```
    <div class="header">
        <h1>HEALTH RISK PREDICTOR Report</h1>
        <p>Generated: {timestamp}</p>
    </div>
    <div class="grid">
        <div class="img-card"><h4>Original Sample</h4><img src="data:image/jpeg;base64,{orig_b64}"></div>
        <div class="img-card"><h4>CV Analysis</h4><img src="data:image/jpeg;base64,{cv_b64}"></div>
    </div>
    <h3>Biomarker Analysis</h3>
    <table>
        <thead><tr><th>Metric</th><th>Risk</th><th>Value</th><th>Reference</th></tr></thead>
        <tbody>{rows}</tbody>
    </table>
    <div class="footer">
        Developed by <strong>M DHIVAGARAN</strong>. AI-generated results for educational screening only.
    </div>
  </body>
  </html>
  """
  return html
class InferenceEngine:
  _instance = None
  def __new__(cls):
    if cls._instance is None:
      cls._instance = super(InferenceEngine, cls).__new__(cls)
      cls.mp_mesh = mp.solutions.face_mesh
      cls.model = cls.mp_mesh.FaceMesh(
        static_image_mode=True,
        max_num_faces=1,
        refine_landmarks=True,
        min_detection_confidence=0.6
      )
    return cls._instance
class BioSignalProcessor:
  def get_roi_lab(image, landmarks, indices, w, h):
    try:
      points = [np.array([int(landmarks[i].x * w), int(landmarks[i].y * h)]) for i in indices]
      x, y, w_box, h_box = cv2.boundingRect(np.array(points))
      if x < 0: x = 0
      if y < 0: y = 0
      if w_box <= 0 or h_box <= 0: return None
      pad = 2
      roi = image[max(0, y-pad):y+h_box+pad, max(0, x-pad):x+w_box+pad]
      if roi.size == 0: return None
      return cv2.cvtColor(roi, cv2.COLOR_RGB2LAB)
    except Exception:
      return None
  def analyze_jaundice(img, lm, w, h) -> HealthMetric:
    lab = BioSignalProcessor.get_roi_lab(img, lm, [33, 7, 163, 144, 153], w, h)
    score = np.mean(lab[:, :, 2]) if lab is not None else 0
    risk, color = "Normal", "#28a745"
    if score > 140: risk, color = "High Risk", "#dc3545"
    elif score > 135: risk, color = "Elevated", "#ffc107"
    return HealthMetric("Liver (Jaundice)", risk, score, "idx", "Sclera yellowing check", color)
  def analyze_anemia(img, lm, w, h) -> HealthMetric:
    lab = BioSignalProcessor.get_roi_lab(img, lm, [145, 153, 154, 155], w, h)
    score = np.mean(lab[:, :, 1]) if lab is not None else 0
```

```
        risk, color = "Normal", "#28a745"
        if score > 0 and score < 132: risk, color = "Possible Anemia", "#dc3545"
        return HealthMetric("Anemia (Iron)", risk, score, "idx", "Conjunctival pallor check", color)
    def analyze_cyanosis(img, lm, w, h) -> HealthMetric:
        lab_lip = BioSignalProcessor.get_roi_lab(img, lm, [78, 191, 80, 81, 82], w, h)
        lab_skin = BioSignalProcessor.get_roi_lab(img, lm, [116, 117, 118], w, h)
        val = 0
        if lab_lip is not None and lab_skin is not None:
            val = np.mean(lab_skin[:, :, 2]) - np.mean(lab_lip[:, :, 2])
        risk, color = "Normal", "#28a745"
        if val > 20: risk, color = "Hypoxia Risk", "#dc3545"
        return HealthMetric("Cyanosis (Oxygen)", risk, val, "ΔB", "Lip vs Skin contrast", color
    def analyze_swelling(lm, w, h) -> HealthMetric:
        try:
            left_ids = [234, 127, 139, 50, 205]
            right_ids = [454, 356, 368, 280, 425]
            left_pts = np.array([[lm[i].x*w, lm[i].y*h] for i in left_ids], dtype=np.int32)
            right_pts = np.array([[lm[i].x*w, lm[i].y*h] for i in right_ids], dtype=np.int32)
            l_area = cv2.contourArea(left_pts)
            r_area = cv2.contourArea(right_pts)
            if r_area == 0: return HealthMetric("Swelling", "Error", 0, "", "", "#000"
            ratio = abs(l_area - r_area) / max(l_area, r_area)
            risk, color = "Normal", "#28a745"
            if ratio > 0.20: risk, color = "Significant Asymmetry", "#dc3545"
            return HealthMetric("Face Swelling", risk, ratio*100, "%Diff", "Cheek volume comparison",
color)
        except:
            return HealthMetric("Swelling", "Error", 0, "", "", "#000")
    def analyze_skin_lesions(img, lm, w, h) -> HealthMetric:
        lab = BioSignalProcessor.get_roi_lab(img, lm, [116, 123, 147, 187, 205, 50], w, h)
        area_max = 0
        if lab is not None:
            l_chan = lab[:, :, 0]
            _, mask = cv2.threshold(l_chan, 80, 255, cv2.THRESH_BINARY_INV)
            cnts, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
            if cnts:
                area_max = max([cv2.contourArea(c) for c in cnts])
        risk, color = "Low Risk", "#28a745"
        if area_max > 100: risk, color = "High Risk", "#dc3545"
        elif area_max > 30: risk, color = "Monitor", "#ffc107"
        return HealthMetric("Skin Lesion (Cancer)", risk, area_max, "px", "ABCD Rule Check", color
def draw_overlay(image, lm_obj, w, h):
    overlay = image.copy()
    lm = lm_obj.landmark
        regions = [
        ([33, 7, 163, 144, 153], (0, 255, 255), "LIVER"),
        ([145, 153, 154, 155], (0, 0, 255), "BLOOD"),
        ([78, 191, 80, 81, 82], (255, 0, 0), "O2"),
    mp.solutions.drawing_utils.draw_landmarks(
        image=overlay, landmark_list=lm_obj,
        connections=mp.solutions.face_mesh.FACEMESH_TESSELATION,
        landmark_drawing_spec=None,
        connection_drawing_spec=mp.solutions.drawing_styles.DrawingSpec(color=(200,200,200),
thickness=1, circle_radius=0
    for ids, col, label in regions:
        try:
            points = [np.array([int(lm[i].x * w), int(lm[i].y * h)]) for i in ids]
            x, y, wb, hb = cv2.boundingRect(np.array(points))
            cv2.rectangle(overlay, (x, y), (x+wb, y+hb), col, 2)
            cv2.putText(overlay, label, (x, y-5), cv2.FONT_HERSHEY_SIMPLEX, 0.4, col, 1)
```

```
        except: pass
    return overlay
def main():
    engine = InferenceEngine()
    st.title("HEALTH RISK PREDICTOR")
    st.markdown("**AUTHOR : DHIVA & ADHAVAN** | *AI-Powered Medical Screening*")
    tab_cam, tab_up = st.tabs(["🎥 Live Camera", "🗁 Upload Photo"])
    img_buffer = None
    with tab_cam:
        cam_val = st.camera_input("Capture Face")
        if cam_val is not None:
            img_buffer = cam_va
    with tab_up:
        up_val = st.file_uploader("Upload Image", type=['jpg','png','jpeg'])
        if up_val is not None:
            img_buffer = up_val
    if img_buffer is not None:
        try:
            raw_pil = Image.open(img_buffer)
            raw_np = np.array(raw_pil)
            # Handle PNG Alpha Channel
            if raw_np.shape[2] == 4:
                raw_np = cv2.cvtColor(raw_np, cv2.COLOR_RGBA2RGB)
            h, w, _ = raw_np.shap
           with st.spinner("Analyzing Biological Signals..."):
                res = engine.model.process(raw_np)
                if res.multi_face_landmarks:
                    lm_obj = res.multi_face_landmarks[0]
                    lm = lm_obj.landmark
                    # 1. Visualization
                    cv_overlay = draw_overlay(raw_np, lm_obj, w, h
                    proc = BioSignalProcessor()
                    metrics = [
                        proc.analyze_jaundice(raw_np, lm, w, h),
                        proc.analyze_anemia(raw_np, lm, w, h),
                        proc.analyze_cyanosis(raw_np, lm, w, h),
                        proc.analyze_swelling(lm, w, h),
                        proc.analyze_skin_lesions(raw_np, lm, w, h)
                    ]
                    c1, c2 = st.columns([1, 1])
                    with c1:
                        st.image(cv_overlay, caption="Computer Vision Analysis", use_container_width=True)
                    with c2:
                        st.subheader("Diagnostic Results")
                        for m in metrics:
                            st.markdown(f"**{m.name}**: <span style='color:{m.color}; font-
weight:bold'>{m.risk}</span>", unsafe_allow_html=True)
                            st.caption(f"Value: {m.value:.1f} | {m.reference}")
                            st.divider()
                    ts = time.strftime("%Y-%m-%d %H:%M:%S")
                    html = generate_medical_report(raw_pil, cv_overlay, metrics, ts)
                    st.download_button("📥 Download Official Report", html, "Health_Report.html",
"text/html")
   else:
                    st.error("No face detected. Please center your face in the camera.")
                     except Exception as e:
            st.error(f"Error processing image: {e}")
if __name__ == "__main__":
    main()
```