Technical Recruitment Challenge 2025

End-to-End Machine Learning Web Application Development
Problem Theme: Skin Lesion Detection

NAME : M.J.ASWIN BHARATHI
REG_NO : 24MIS1080

1. Estimated Tech Stack

Tech Stack

- Frontend:
    - Flask Jinja templates (HTML, CSS, minimal JS)
    - Inline CSS (Bootstrap/Tailwind optional if you want more polish)
- Backend:
    - Flask (Python web framework)
    - SQLite (local DB for users + upload history)
- ML/DL Framework:
    - TensorFlow/Keras (**`MobileNetV2`** for transfer learning)

    - NumPy, Matplotlib for data handling & visualization

- Deployment Plan:
    - Option 1: Local run (Python + requirements.txt)
    - Option 2: Cloud
        - Free: Render, Railway, or Heroku
        - Containerized: Dockerfile → deploy on AWS/GCP/Azure/Streamlit Cloud

---

2. Final Report

- Project Title & Description
  Skin Lesion Detection Web App using MobileNetV2. Upload an image → classify as benign/malignant.

- Tech Stack (as above)

- System Architecture
  Show a flow diagram (user → Flask → Model → Prediction → Visualization → DB).

- Features Implemented
    - User
    - Confidence & summauthentication

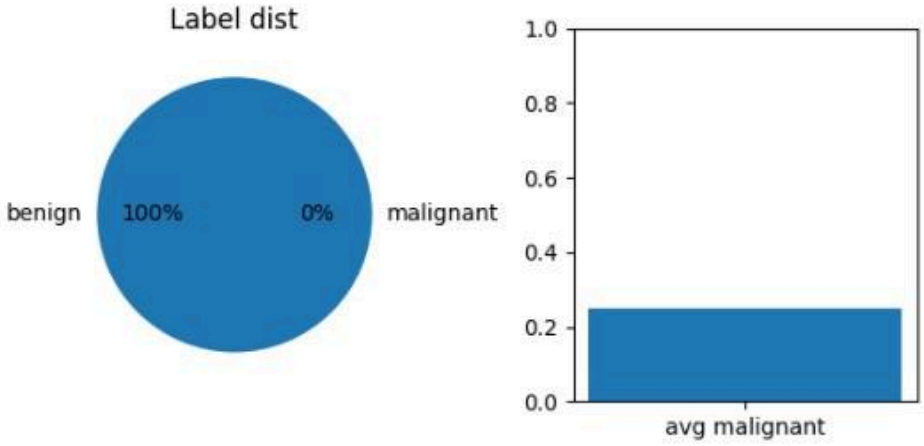
    - Image upload + predictions
    - History of results
      ary charts
    - Styled UI
- ML Model
    - Transfer learning using MobileNetV2
    - Input: 224x224 images
    - Binary classification (sigmoid)
- **Inputs:**

- **Screenshots Of Outputs:**

Your Upload History

- 2025-09-15T12:49:03.252759 - benign (0.25)

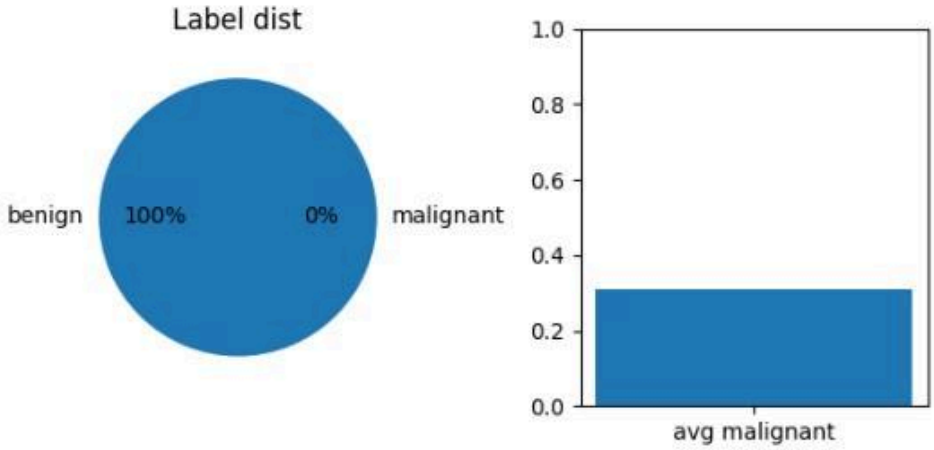
Summary



Your Upload History

- 2025-09-15T12:53:32.581081 - benign (0.31)

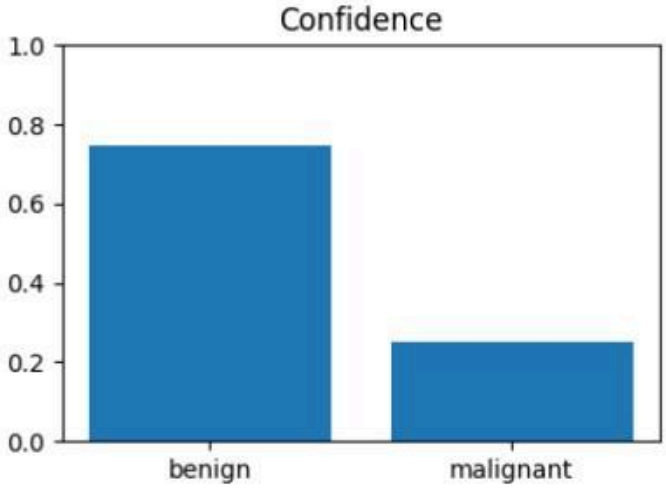
Summary



Home

Prediction Result

Label: benign | **Score:** 0.25

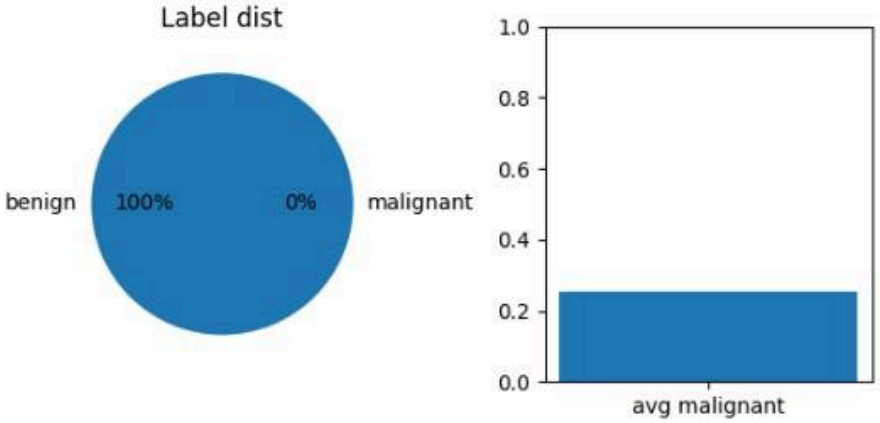


Back to history

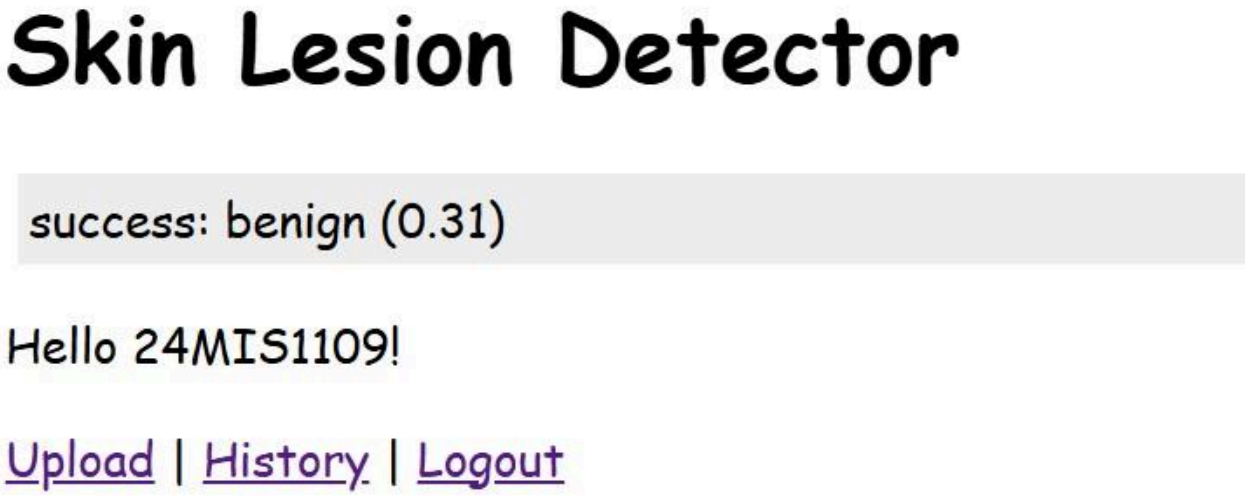
Your Upload History

- 2025-09-15T12:57:07.263961 - benign (0.25)

Summary



Home



- **Future Improvements**
    - **Deploy on cloud**
    - **Grad-CAM explainability**
    - **Dark mode**

3. GitHub Repository Link (source code with README):
https://github.com/ASWIN1234569867/Skin-Lesion-Detection-.git
4. Live Demo Link (if deployed). If not deployed, provide local setup instructions.
Local Setup:
 # Run app
python app.py
 # Open in browser
http://127.0.0.1:5000/