

Noor.AI – Skin Analysis Feature Development Report

1. Introduction

The Skin Analysis feature of Noor.AI uses machine learning to classify skin types from user images. It is built using Python and TensorFlow, leveraging OpenCV for image preprocessing and trained on a curated Kaggle dataset. Development and version control were managed with VS Code and Git/GitHub. For the user interface, assistance was provided by Claude and Cursor.ai. This module was developed and rigorously tested independently prior to integration.

2. Tools & Technologies Used

Tool	Purpose
Python	Core programming for the project
Teachable Machine	Creating and training the skin model
OpenCV	Handling image input and preprocessing
Git & GitHub	Version control and collaboration
VS Code	Writing and editing the code
Kaggle	Source of the dataset
Claude	Assistance with building the user interface
Cursor.ai	AI-powered support for UI development

3. Dataset & Preprocessing

- Used a curated dataset of facial images that I downloaded from Kaggle annotated with skin types (dry, oily, combination, normal).
- Applied normalization to prepare images for model input.
- Dataset split into 80% training, 10% validation, and 10% testing.

4. Model Architecture (TensorFlow)

- I used Google's Teachable Machine to create the skin type classification model.
- It allowed me to upload images of different skin types and train a model to recognize them.
- The model was trained and reached an accuracy of around 85%.
- After training, I downloaded the model in a format that could be used in my Noor.AI project.
- This helped me add a feature where the system can analyze an image and guess the user's skin type.

5. Feature Workflow

- User uploads an image through the UI.
- Backend preprocesses the image.
- Model predicts skin type.
- Output is returned to the frontend and stored for recommendation.

6. Challenges Faced

- Making sure the dataset had good-quality images and a fair number of examples for each skin type.
- Picking the best settings (like learning rate and batch size) to help the model learn better.
- Keeping the Git commit history clean by avoiding repeated or unnecessary commits.

7. Future Improvements

- Add more skin type categories like sensitive and acne-prone to make the model more helpful.
- Use a pre-trained model (transfer learning) to improve accuracy without starting from scratch.

8. Screenshots

Steps Taken to Develop the Skin Analysis AI Model:

Step 1: Dataset Collection and Model Training

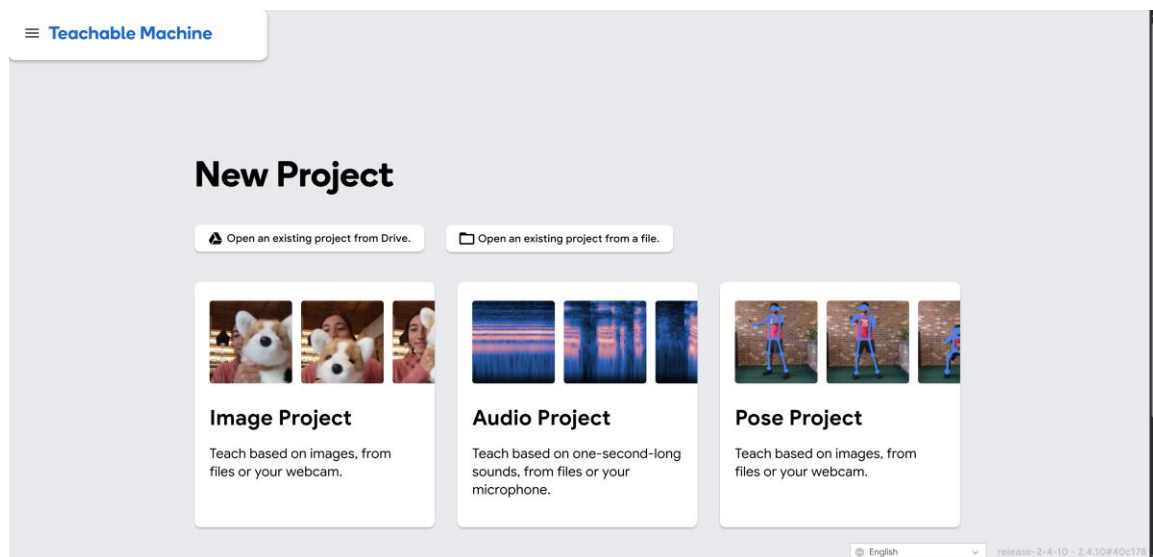
I began by downloading a skin image dataset from Kaggle. To build the AI model, I used Teachable Machine with the standard image classification setup. I created three classes: Dry Skin, Normal Skin, and Oily Skin, and uploaded the corresponding images into each class. After organizing the dataset, I trained the model, which completed in approximately 15 minutes. Once training was done, I exported the model, which generated two essential JSON files required for integration.

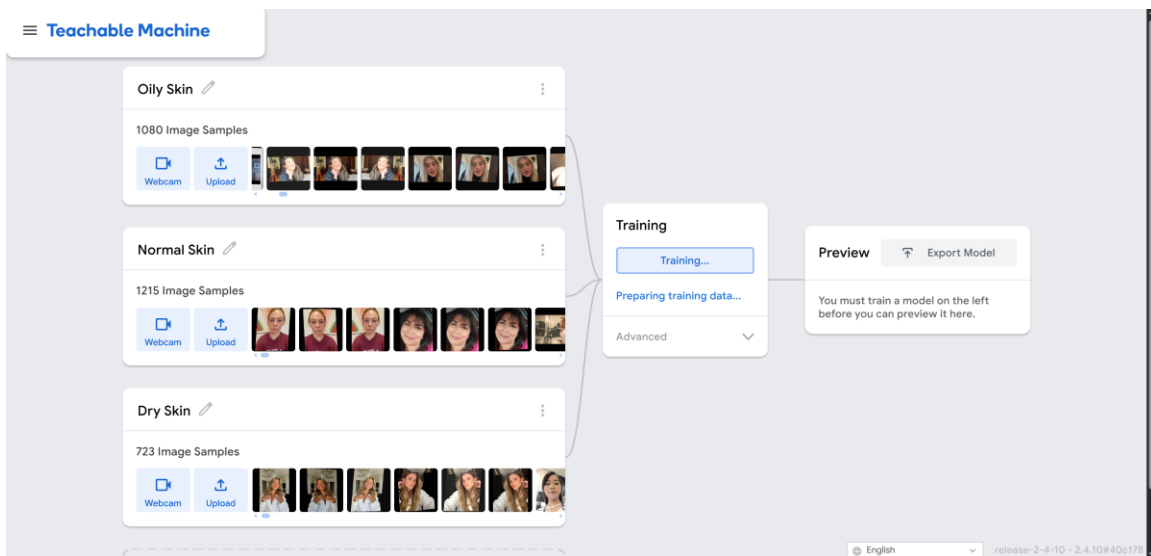
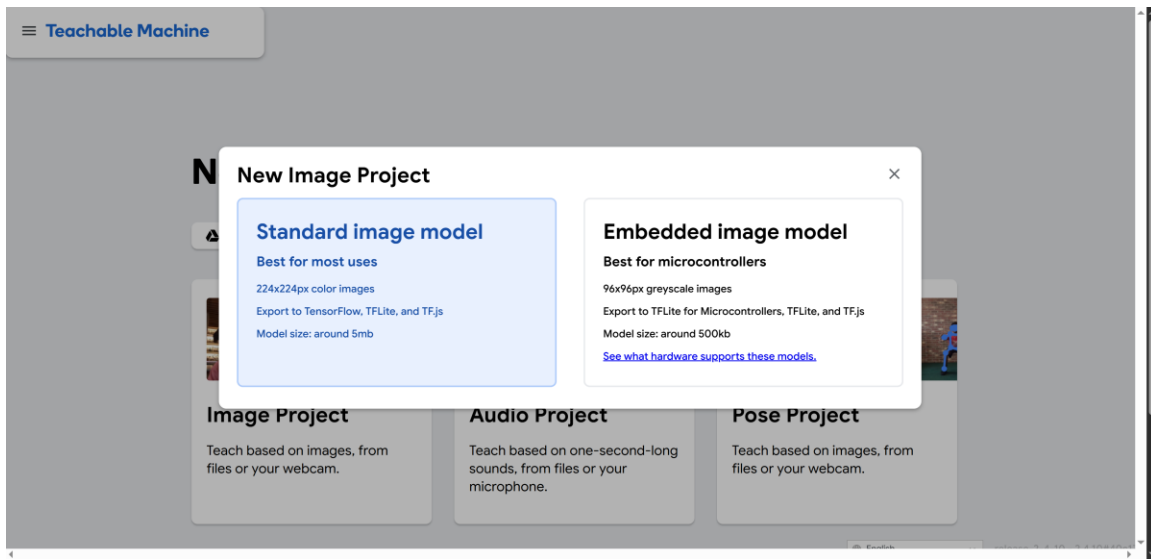
Step 2: Model Integration and UI Development

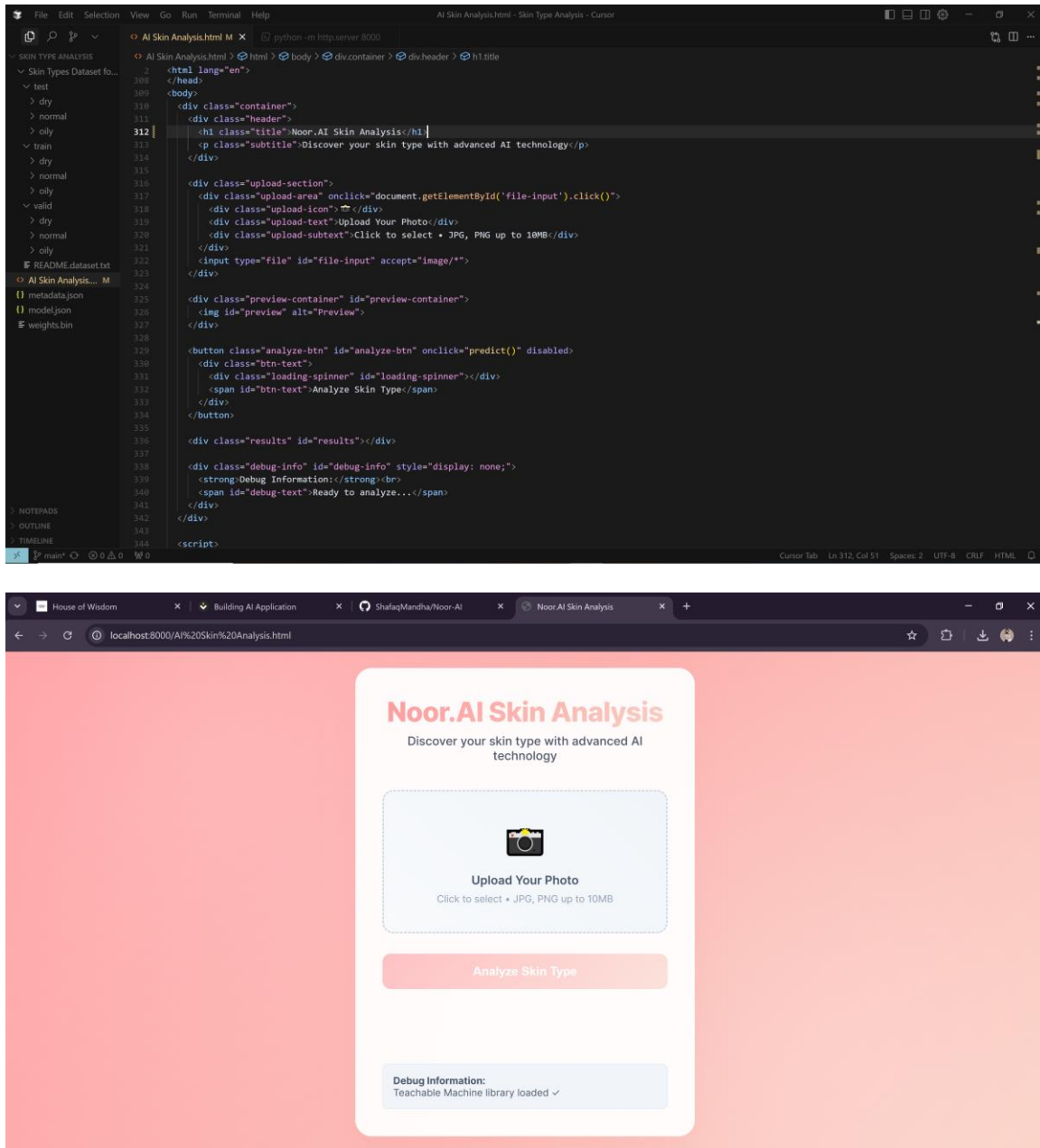
After obtaining the model files, I moved on to integrate the model into my application. I began building the user interface (UI) with assistance from Claude and Cursor.ai, focusing on creating a smooth and user-friendly experience that could interact with the AI model effectively.

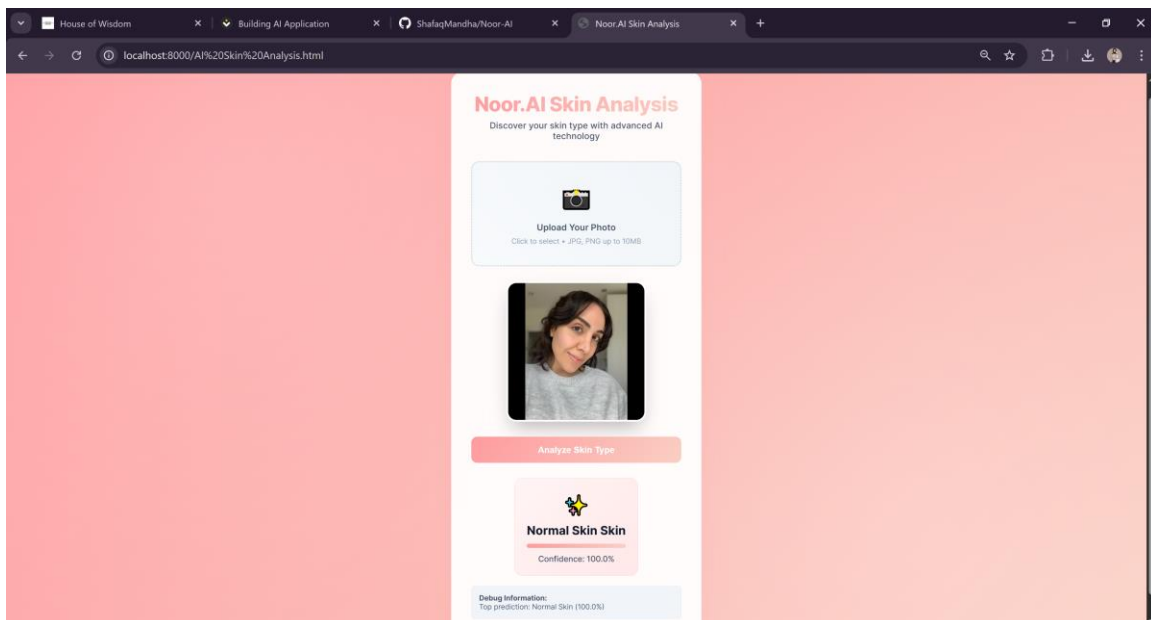
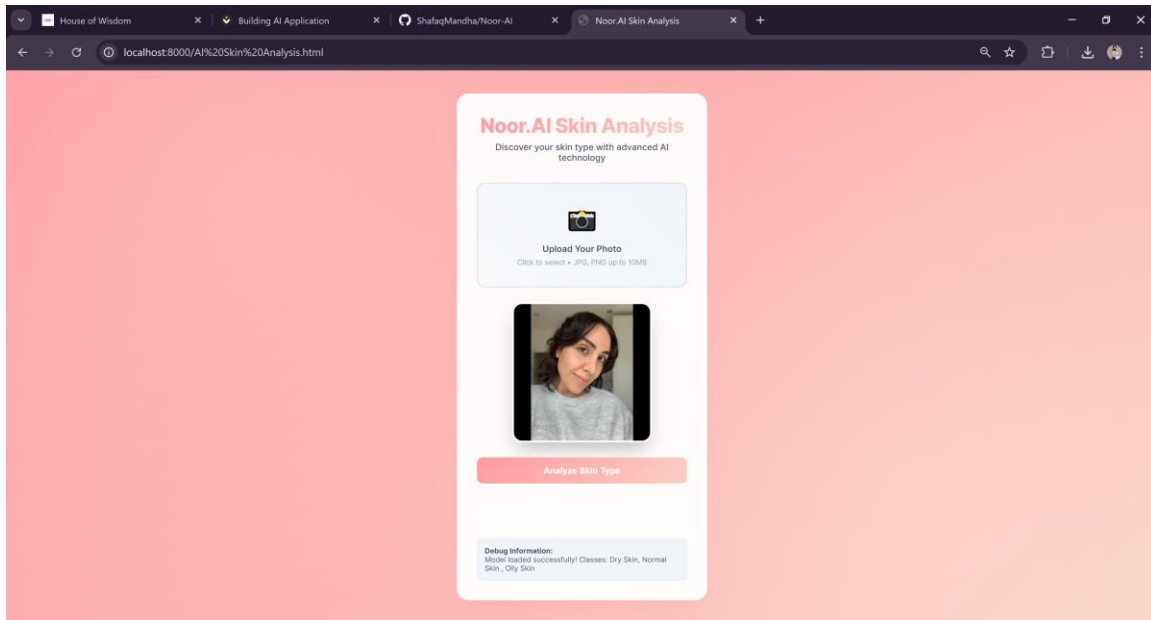
Step 3: Testing and Validation

Finally, I conducted thorough testing of the integrated AI model to ensure its accuracy and reliability in identifying the correct skin type based on user input. I made adjustments where needed to improve performance and usability.









9. Conclusion

This standalone module has been successfully built, tested, and documented. It is now ready for integration with other Noor.AI components.