Project Title: Melatone: Smart Skin Tone Advisor

Member(s): Bui Cao Dong Nghi - Tran Nguyen Hanh Nguyen

Introduction

Melatone is a real-time computer vision application that analyzes facial skin tone and recommends personalized color palettes tailored to a user's complexion. The system integrates face detection, skin region segmentation, deep learning-based tone classification, and aesthetic color theory within a user-friendly Streamlit interface. Users can either upload a photo or take one via webcam; the application then predicts their skin tone category (Light, Medium, or Dark) and generates a curated palette supporting fashion, design, or cosmetics choices.

We selected this project because of our shared interest in both AI applications and human-centered design. Melatone bridges technical model deployment with visual creativity, aligning with our goal to develop AI tools that are not only functional but also intuitive and accessible for everyday users. We recognize that personal color analysis services can often be prohibitively expensive and out of reach for many people. With Melatone, we aspire to provide a free, real-time solution that makes personalized color recommendations more inclusive and accessible, giving everyone the opportunity to discover and embrace their

Method

The overall architecture of Melatone consists of five main components:

1. **Face and Skin Detection:**

The development process started with face and skin detection using OpenCV's Haar Cascade classifier, which allowed us to efficiently isolate the facial region in uploaded images and live webcam feeds. To extract the skin region, we employed a classical computer vision pipeline. First, we converted the image to the HSV color space and then applied a threshold using pre-defined HSV ranges corresponding to typical skin tones.

The resulting binary mask was refined using morphological operations, specifically opening and closing, and smoothed with Gaussian blur. Next, we used bitwise masking to extract the skin pixels and cropped the region of interest using heuristic coordinates targeting the cheek and forehead. This approach enabled effective skin segmentation without needing a learning-based segmentation model, keeping the process lightweight and efficient.

2. **Initial ML Approach with HOG + KNN/XGBoost:**

Initially, we explored conventional machine learning approaches with Histogram of Oriented Gradients (HOG) feature extraction paired with K-Nearest Neighbors and XGBoost classifiers. These models, however, demonstrated limited robustness to skin tone variability and lighting conditions, achieving only 30–71% accuracy. These limitations validated the need for a more advanced deep learning approach.

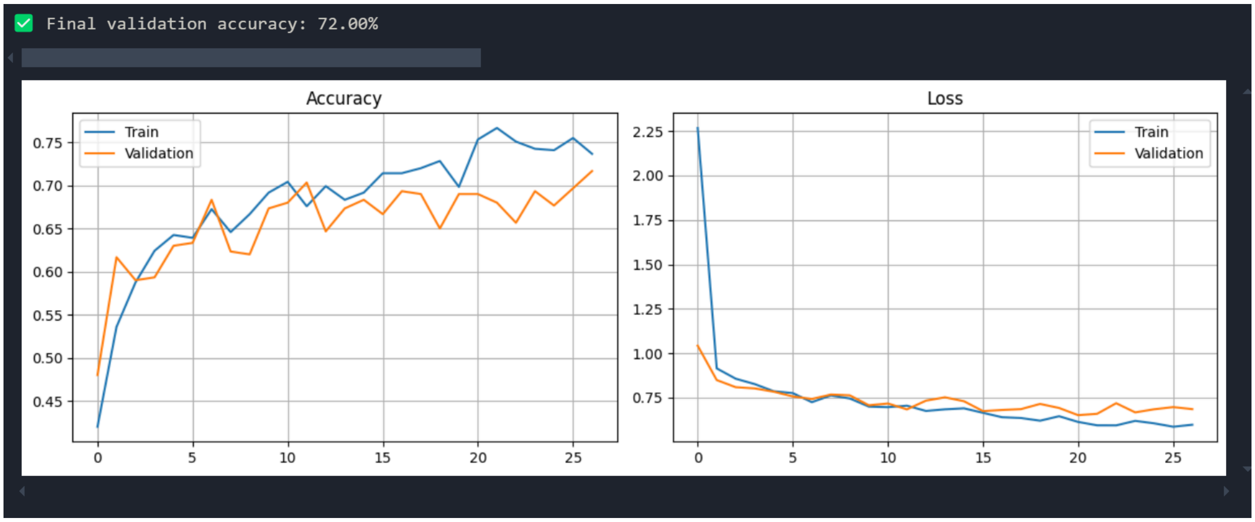
3. **Deep Learning Approach**

We implemented a convolutional neural network (CNN)-based classifier trained on 1500 images labeled as Light, Medium, or Dark to improve accuracy and generalization. Throughout development, we tested multiple architectures to compare their performance. Our initial training with VGG16 yielded relatively low accuracy and poor generalization.

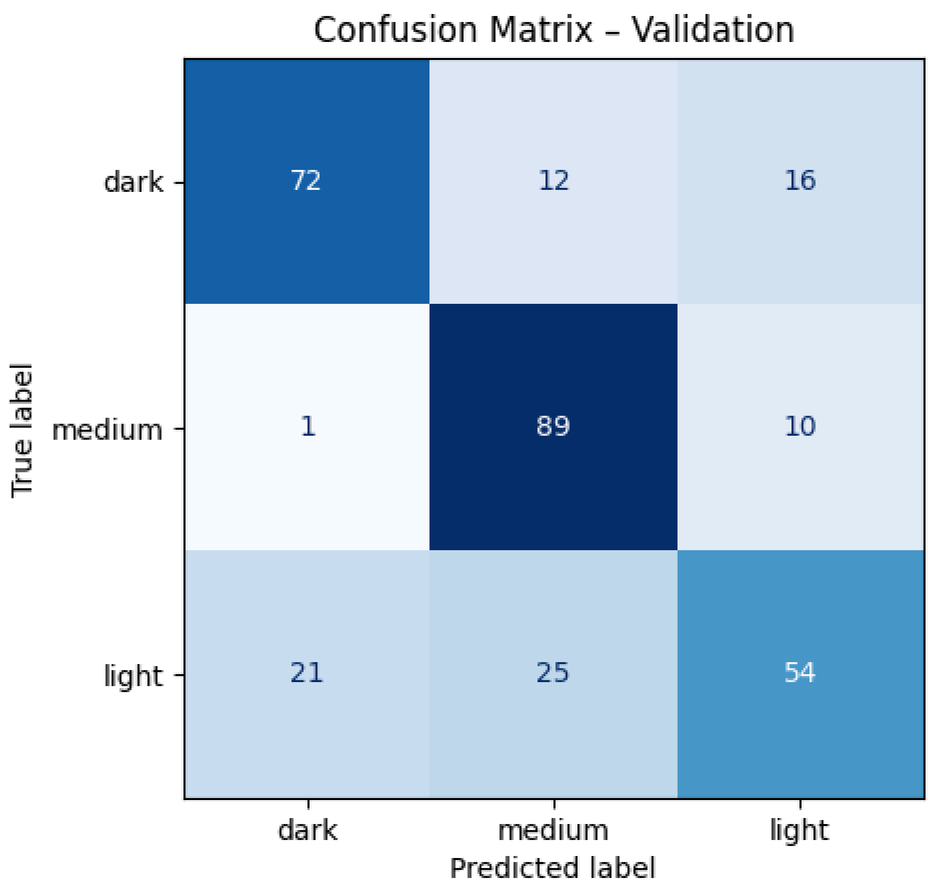
```
# Evaluate the Model
val_loss, val_acc = model.evaluate(val_generator, verbose=1)
print(f"Validation Accuracy: {val_acc:.2%}")

10/10 ━━━━━━━━━━━━━━━━━━━━ 6s 625ms/step - accuracy: 0.4824 - loss: 0.9985
Validation Accuracy: 49.67%
```

Feature 1: Result on VGG16

Feature 2: VGG16 Model Loss and Model Accuracy



Feature 3: Second try on VGG16, Early Stopping Applied



Feature 4: Confusion Matrix of VGG16, second attempt

We then moved to EfficientNet, which showed notable improvement in validation accuracy. Ultimately, we built and fine-tuned a custom EfficientNet architecture using PyTorch, which achieved the best performance, reaching approximately 75.3% validation accuracy. Our team collaboratively designed, trained, and evaluated this final model, with experimentation informed by research literature and real-world testing.

We designed and developed the majority of the system architecture, data processing pipeline, and model implementation. This includes face detection, skin region cropping, white balance correction, color palette recommendation logic, and initial model training using VGG16.

AI tools, including ChatGPT, were consulted primarily during the model refinement and performance tuning phase. For example, after observing suboptimal results with VGG16, we explored alternatives suggested by AI, such as switching to EfficientNet, which ultimately provided better generalization. AI assistance was also sought for specific optimization techniques, including the use of the Adam optimizer, early stopping, and dropout for regularization.

In summary, the product's foundation, data preparation, and logic flow were built by us, while AI tools were used selectively to guide architectural choices and training improvements.

4. **Color Palette Recommendation:**

The color palette recommendation was developed independently by curating HEX-based palettes aligned with traditional seasonal color theory and then adapted for Southeast Asian skin tones. Each skin tone class is hard-coded to its palette, ensuring consistency and explainability.

List of HEX color palettes for each tone category

Feature 5: Color Palette for Light Skin Tone

Swatch #	HEX Code	Suggested Name
1	#FDF3E1	Porcelain Glow
2	#FBE8D3	Ivory Blush
3	#C6895E	Toasted Caramel
4	#F3CDA4	Honey Beige
5	#F9E5AC	Buttercream

6	#F5DADA	Rose Petal
7	#F3B1A6	Blush Coral
8	#F9D269	Golden Peach
9	#F7E2A1	Pale Marigold
10	#CBE1EF	Powder Blue
11	#E7D3ED	Lavender Mist
12	#F3EFEA	Linen
13	#F8E8D5	Vanilla Cream
14	#E6E3D7	Soft Olive

Feature 6 Color Palette for Medium Skin Tone

Swatch #	HEX Code	Suggested Name
1	#f5f2eb	Vanilla Cream
2	#d7cfc5	Sand Beige
3	#c8c5c0	Silver Ash
4	#798472	Olive Dust
5	#758f87	Sage Smoke

6	#f6dbd5	Blush Petal
7	#f9e4a0	Lemon Drop
8	#f5d7a1	Pale Gold
9	#d0a78e	Mellow Apricot
10	#9e6045	Clay Terracotta
11	#b1775f	Sunset Brown
12	#a3d2b3	Seafoam Green
13	#c2e5e0	Aqua Mist
14	#9daecf	Dusty Periwinkle

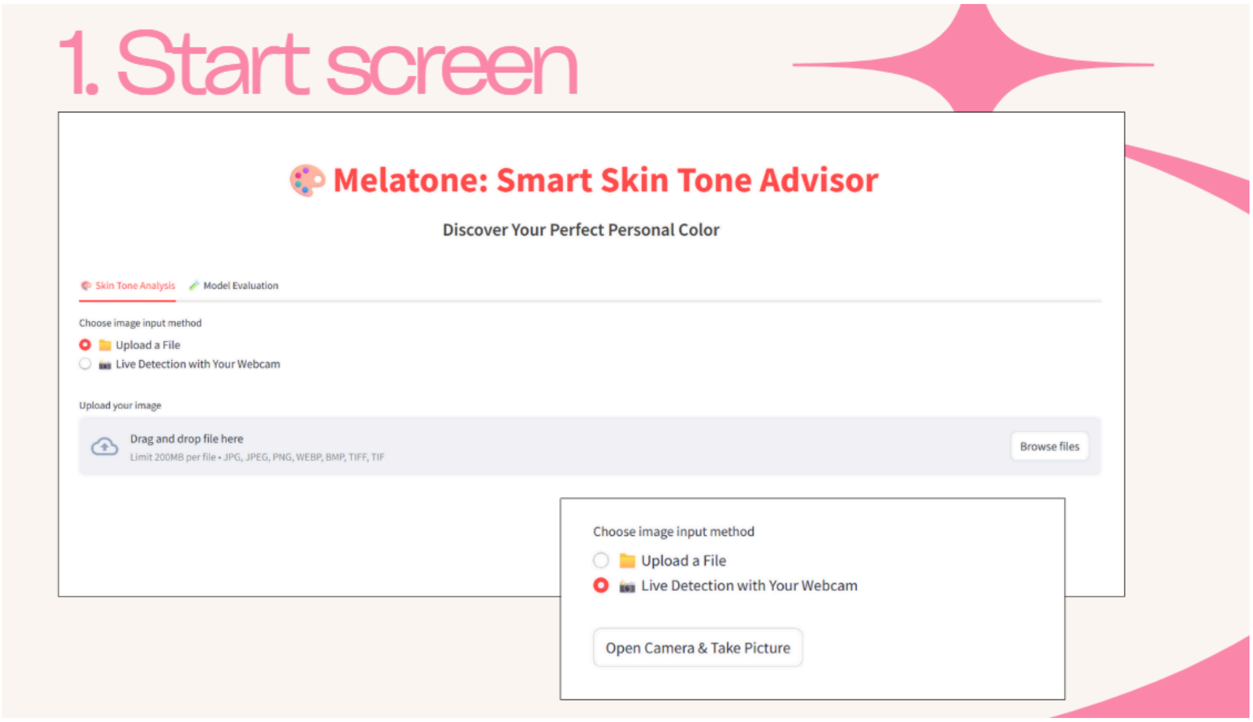
Feature 7 Color Palette for Dark Skin Tone

Swatch #	HEX Code	Suggested Name
1	#2b2b2b	Charcoal Black
2	#3c322d	Mocha Brown
3	#4d4540	Ash Taupe
4	#7b4e3a	Burnt Cocoa
5	#8b634c	Warm Walnut
6	#585e3d	Olive Bark

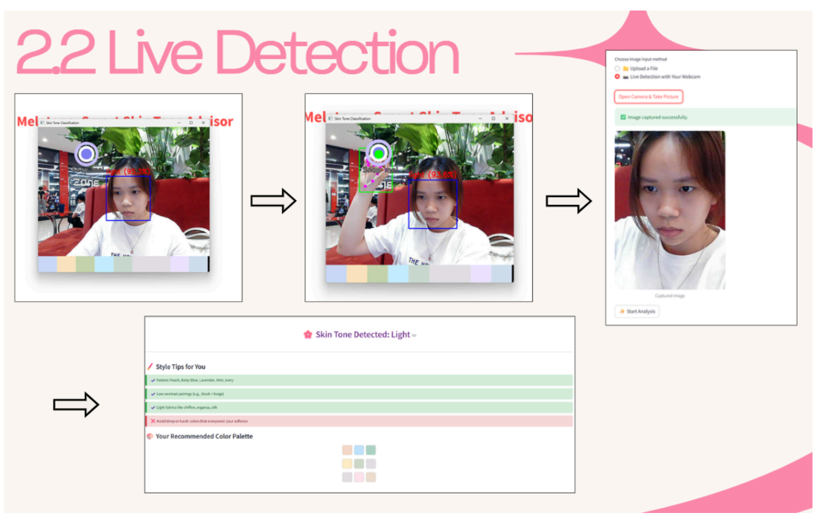
7	#14564a	Teal Forest
8	#6a264d	Plum Wine
9	#902e3b	Ruby Clay
10	#e1a11c	Golden Mustard
11	#b7983b	Antique Brass
12	#526781	Steel Blue
13	#304344	Charred Green
14	#787878	Urban Slate
15	#1c1c1c	Deep Obsidian

5. **User Interface (UI):**

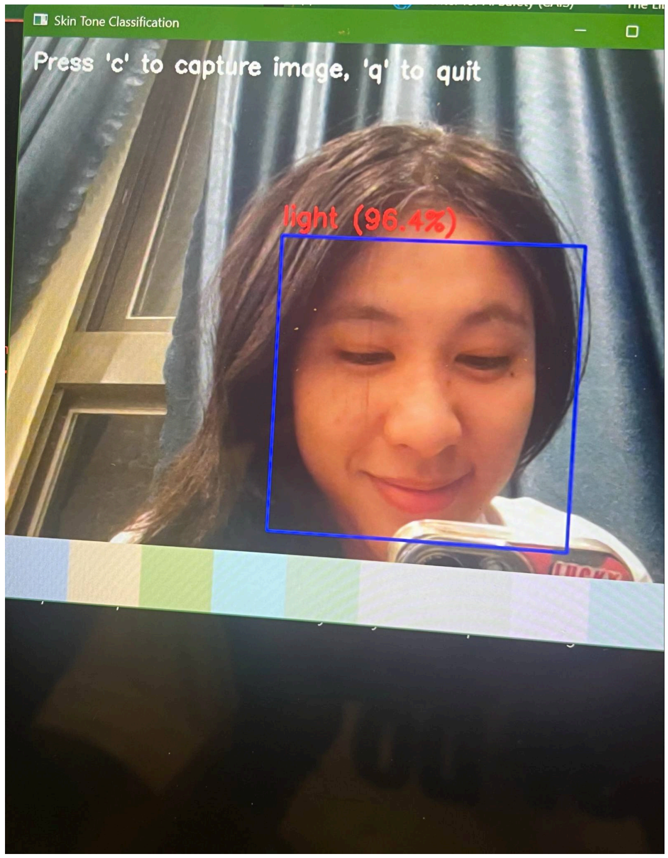
The user interface was developed using Streamlit, enabling input from webcams and file uploads. Initially, we incorporated MediaPipe hand gesture recognition to trigger image capture. However, based on user testing feedback, we streamlined this process by switching to a keypress ('c'), which enhanced usability across various devices. The interface displays the prediction results and the suggested color palette preview. Our team developed this end-to-end system, which covers CV preprocessing, classification, palette rendering, and UI integration, iteratively and collaboratively.



Feature 8: Streamlit UI



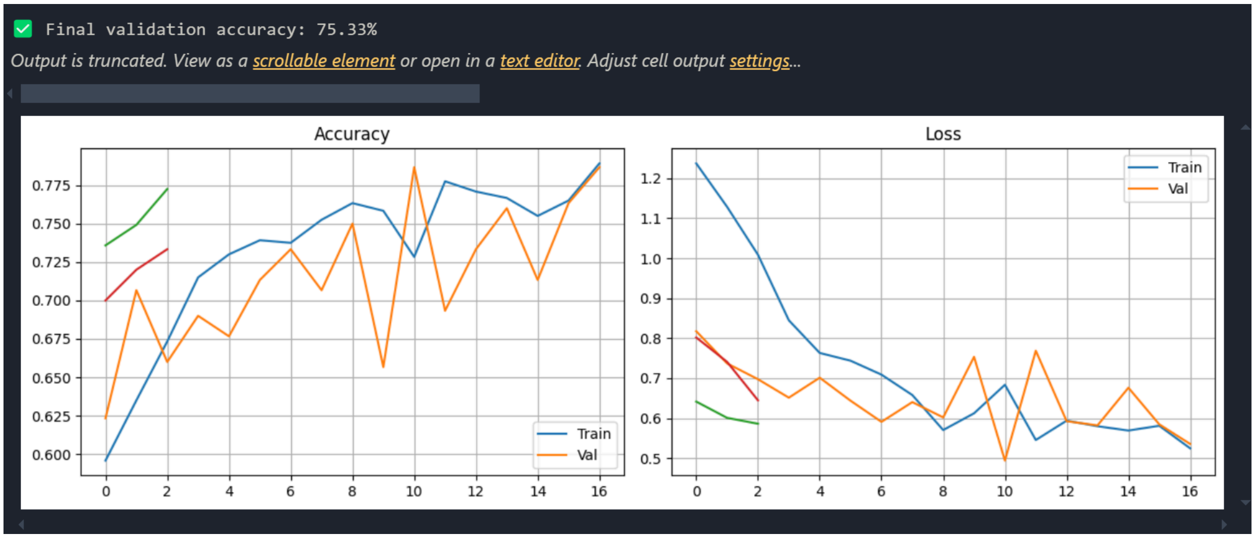
Feature 9: Live Detection using Hand Gesture



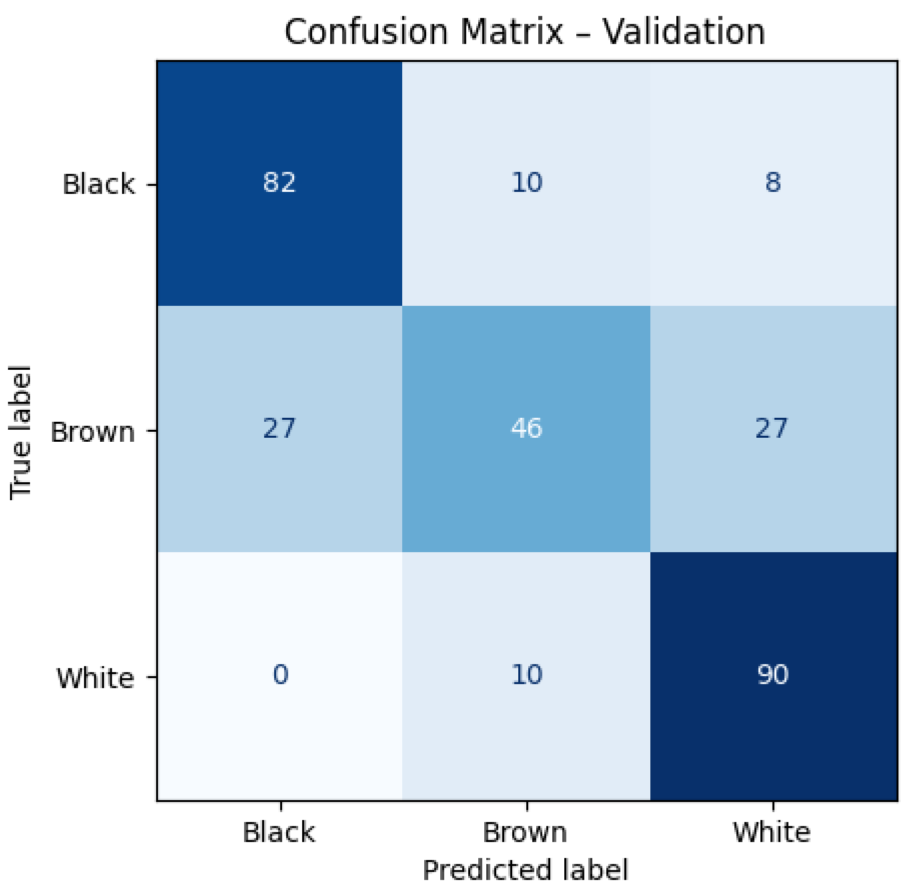
Feature 10: Eliminate the Hand Gesture

Result and Discussion

We used a dataset of 1500 facial skin images from [Kaggle](), each labeled into one of three tone classes: Light, Medium, or Dark. Training was conducted on a local GPU and the final system was tested using both file-uploaded images and webcam-captured photos on a standard laptop. The classifier was trained for 15 epochs using the Adam optimizer in PyTorch, with early stopping and accuracy tracking. We evaluated three CNN architecturesVGG16, EfficientNetV2S, and our custom EfficientNet-based model. While VGG16 yielded relatively poor results and struggled with generalization. Our final EfficientNet-based model outperformed both, achieving a validation accuracy of 75.3%.



Feature 11: EfficientNet Training Result



Feature 12: Confusion Matrix of EfficientNet

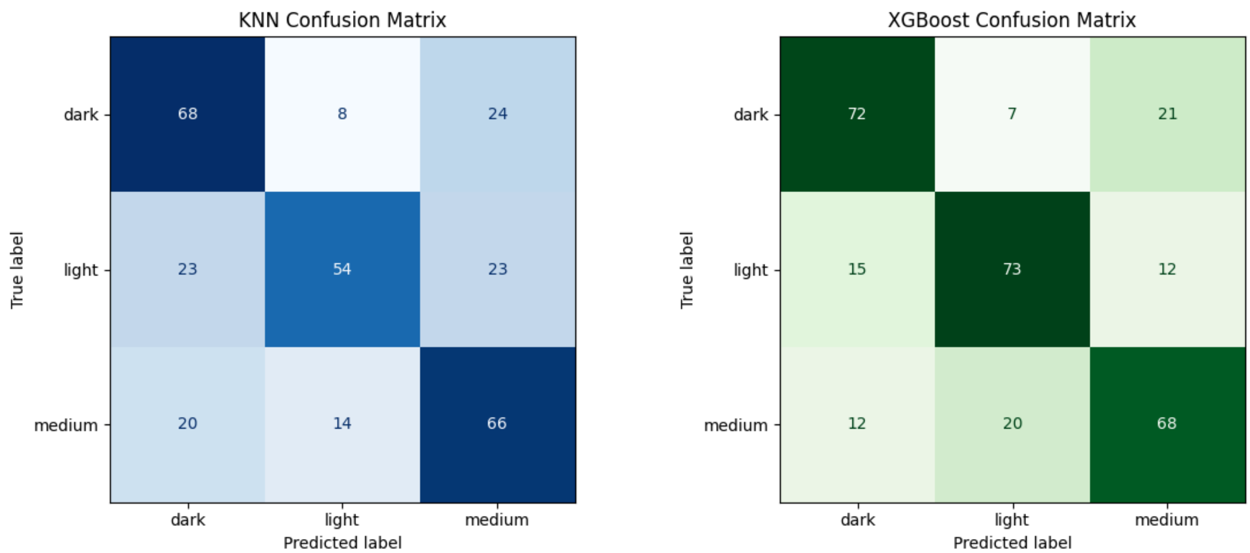
For comparison, conventional ML approaches using HOG features followed by KNN and XGBoost classifiers achieved 63% and 71% accuracy, respectively. Although XGBoost was the stronger of the two classical models, both underperformed relative to deep learning due to their inability to handle color variance and facial region complexity. All experiments confirmed that skin tone predictions made by our EfficientNet classifier aligned well with visual inspection.

```
=== KNN Classification Report ===
              precision    recall  f1-score   support

        dark       0.61      0.68      0.64       100
       light       0.71      0.54      0.61       100
      medium       0.58      0.66      0.62       100

    accuracy                           0.63       300
   macro avg       0.64      0.63      0.63       300
weighted avg       0.64      0.63      0.63       300
```

Feature 13: HOG + KNN

```
=== XGBoost Classification Report ===
              precision    recall  f1-score   support

        dark       0.73      0.72      0.72       100
       light       0.73      0.73      0.73       100
      medium       0.67      0.68      0.68       100

    accuracy                           0.71       300
   macro avg       0.71      0.71      0.71       300
weighted avg       0.71      0.71      0.71       300
```

Feature 14: HOG + XGBoost



Feature 15: Confusion Matrix of KNN and XGBoost

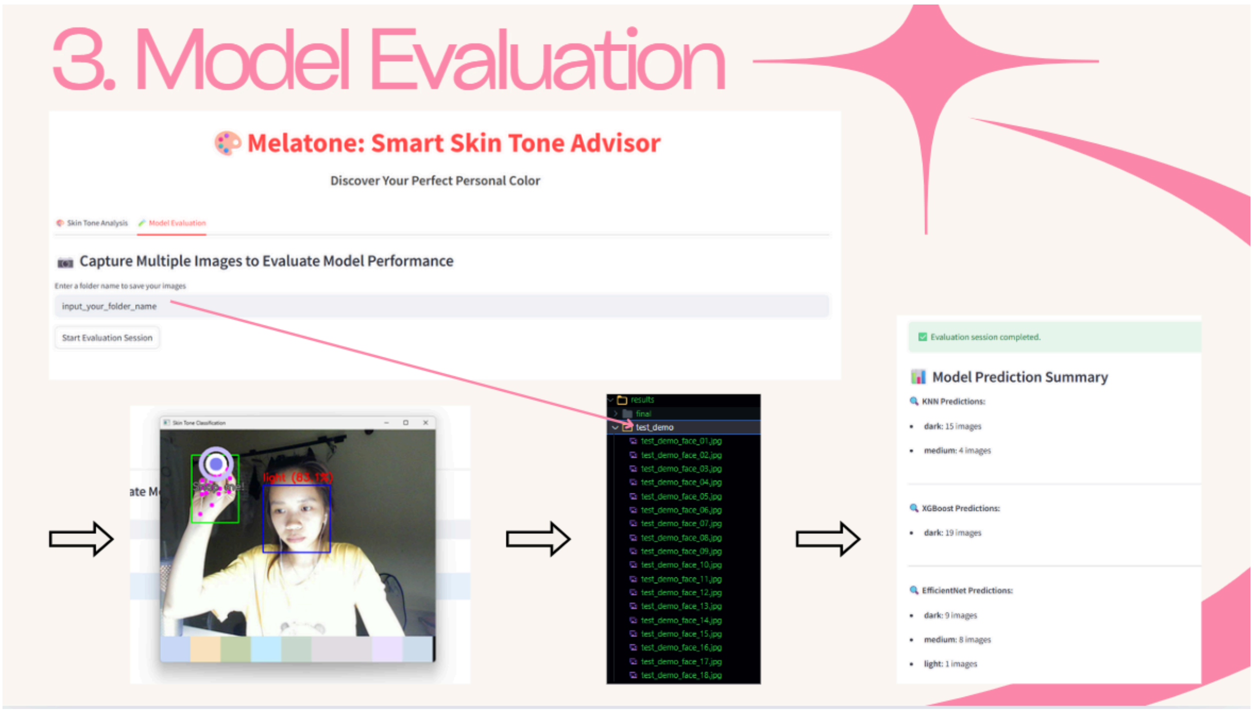
The final application can generate results in real time with an average performance of 2–3 frames per second. The color palette mapping logic was validated through predefined swatches and manual inspection. Regarding per-class performance, the EfficientNet model showed stronger performance in medium and dark tones. At the same time, the Light class had slightly reduced accuracy, likely due to underrepresentation in the dataset. Confusion matrix analysis supported these observations, showing higher false positives in the Light category.

While our current three-class model performs well for general use, further improvements, such as tone diversification and more granular skin segmentation, would enhance fairness and precision. Additionally, future versions of Melatone could adopt more advanced face detection frameworks like YOLOv8 to improve robustness, especially under challenging lighting conditions or for diverse facial angles.

Model Evaluation Session

To better compare the performance of different classification models, we developed a post-processing function that allows for batch image evaluation and displays predictions from KNN, XGBoost, and EfficientNet classifiers. Users can run a Streamlit-based interface that captures multiple webcam images and stores them in a session folder.

After capturing, the system generates a Model Prediction Summary, which outlines the number of images each model classifies into Light, Medium, or Dark categories. This pipeline allows for intuitive visual comparisons and error spotting between traditional ML and deep learning models.



Feature 16: Model Evaluation

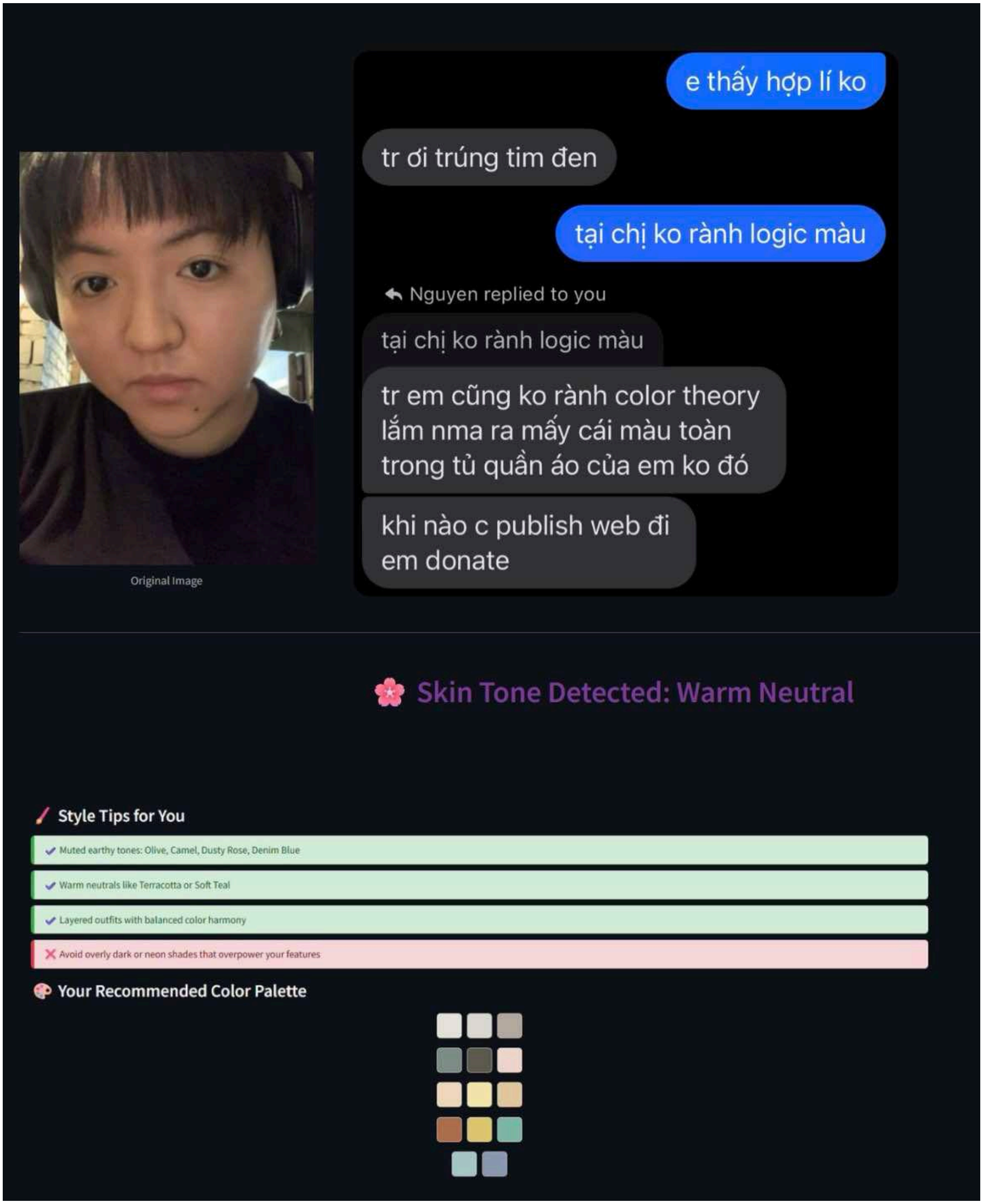
This tool was crucial for identifying over-prediction in certain classes (e.g., XGBoost's bias toward Dark), and verifying EfficientNet's more balanced output.

To improve the consistency of skin tone classification under varying lighting conditions, we integrated a white balance correction step into the preprocessing pipeline. Specifically, we applied a gray-world white balance algorithm to each captured image before cropping and classification. This adjustment normalizes the color channels by balancing the average intensity of red, green, and blue pixels across the image. Without this correction, images captured under warm indoor lighting or cool outdoor lighting could introduce misleading color tones, potentially affecting the performance of all three models. We significantly reduced lighting-induced variance by applying white balance correction at the input stage in post_processing.py and ensured more reliable tone predictions across sessions.

Feedback from Users

In the early version of our application, the skin tone prediction result was directly displayed as "Light", "Medium", or "Dark", reflecting the classifier's internal label categories. However, through user feedback, particularly from Vietnamese users, we realized that displaying the real skin tone can be culturally insensitive or uncomfortable in some contexts. To address this, we introduced a display mapping dictionary: "Dark" → "Golden Earth", "Medium" → "Warm Neutral", and "Light" → "Soft Radiant". This change preserved the internal classification logic while improving inclusivity and user acceptance. The adjustment required only a UI-level mapping and did not affect the model architecture or prediction pipeline.

Feature 16: Skin Tone Detected: Soft Radiant — and she agreed!



Feature 17: Even users unfamiliar with color theory found our recommendations matched their wardrobe.

Challenges

Our project presents valuable opportunities for improvement in several areas. Our dataset consists of only 1,500 images, limiting our representation across various ethnicities, skin tones, and lighting conditions. Expanding this dataset can enhance the model's fairness and generalization capabilities. Collecting and annotating high-quality skin tone datasets also offers a chance to prioritize privacy, informed consent, and ethical data use. Enhancing our approach in these areas will strengthen the integrity of our work.

Additionally, we currently use fixed, rule-based palettes for color recommendations. By incorporating more contextual factors, such as clothing type and lighting conditions, we can create a more user-tailored experience. Lastly, the varying performance of classification models highlights the importance of consistent evaluation methods. Our post-processing function can be refined to facilitate clearer and more effective evaluations, driving us toward better outcomes.

Future Improvement

We have identified key areas for improvement that will significantly enhance the accuracy and inclusivity of our system. Our model classifies skin tones into Light, Medium, and Dark. However, the reality is that skin tones are much more diverse. We are committed to incorporating more specific categories like Fair, Tan, Olive, and Deep Brown to deliver more precise recommendations. Furthermore, we will replace the Haar Cascade face detector with a YOLO-based framework to achieve superior detection performance across various conditions. By implementing automatic white balance correction, we will effectively eliminate classification errors caused by lighting variations. Our plans also include the development of a dynamic recommendation engine that takes into account color contrast and real-time user feedback, ensuring a tailored experience. Finally, we will enhance the post-processing module to feature confidence score visualization and comprehensive error analysis, making it a powerful tool for refining future versions of Melatone.

Conclusion

In this project, we designed and implemented a complete computer vision pipeline that detects a user's skin tone and recommends a personalized color palette. From early trials with HOG-based machine learning models to deploying a custom EfficientNet classifier and a Streamlit-powered UI, each component was iteratively built and refined. Feedback-informed changes, such as simplifying gesture capture, significantly improved the user experience. Melatone highlights the potential of AI in personal aesthetics and fashion-tech, especially for underrepresented tones such as Southeast Asian skin.

Compared to the initial proposal titled *Smart Outfit Advisor*, the final implementation, named *Melatone,* retained the core idea of real-time skin tone detection and personal color recommendation. However, the system evolved to match user expectations and feasibility constraints better. While the initial goal included gesture-based capture and outfit suggestions, user feedback led us to adopt a simpler keypress-based capture method, and the scope was refined to seasonal color palettes rather than clothing advice. Technically, the project shifted from a rule-based classification to a more robust CNN model trained on a real dataset, which improved prediction accuracy and model generalization. These changes reflect the importance of iterative feedback and practical design decisions in applied computer vision development.

References

Skmbatha. (2021). MEng Skin-tone estimation using VGG-16 [GitHub repository]. GitHub. https://github.com/skmbatha/MEng-Skin-tone-estimation-using-VGG-16

ChenglongMa. (2021). SkinToneClassifier [GitHub repository]. GitHub. https://github.com/ChenglongMa/SkinToneClassifier

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems (Vol. 32). https://pytorch.org/

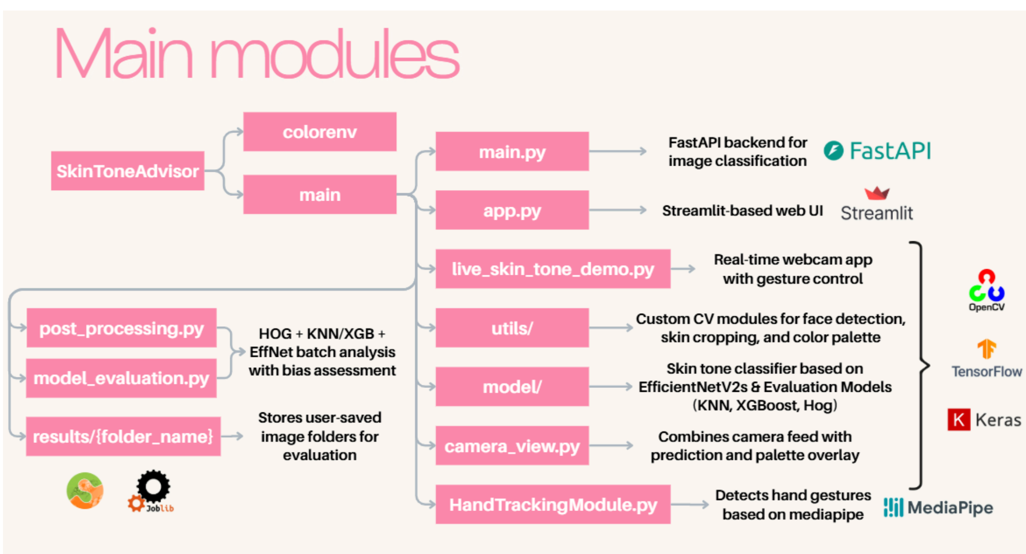
Streamlit Inc. (2024). Streamlit Documentation. https://docs.streamlit.io/

Sasagawa, Y., & Matsui, T. (2010). Real-time face parts recognition using haar-like features and adaboost. Proceedings of the International Conference on Artificial Life and Robotics (ICAROB), 25(6), 1–4. https://alife-robotics.co.jp/members2010/icarob/Papers/OS25/OS25-6.pdf

usamarana. (n.d.). Skin Tone Classification Dataset [Data set]. Kaggle. https://www.kaggle.com/datasets/usamarana/skin-tone-classification-dataset

Appendix

Code base directory structure (see Canva diagram)



Feature 19: Workflow

Link to Canva slide:
https://www.canva.com/design/DAGnJGpdfRk/SI9XagBAbFGcVVkjacWE1g/edit