Group members:
Anza Malik (FA21-BCS-037)
Romysa Siddiqui (FA21-BCS-069) Course:
Topics in Computer Science 2
Semester Project Report

Project Title

Seasonal Skin Palette Analysis: A Machine Learning Approach to Personal Color Classification.

Introduction

Seasonal Color Analysis (SCA) is a method used to determine the most flattering color palettes for individuals based on their natural features—specifically, eye color, hair color, and skin tone. The goal of this project is to develop a machine learning model that classifies individuals into one of four seasonal color categories: **Spring, Summer, Autumn, or Winter**. This classification will provide users with a personalized color palette that compliments their natural features.

Learning Outcomes:

- Develop a machine learning model that classifies users into seasonal color categories.
- Design a system that prioritizes vein color for more accurate undertone detection.
- Create a small but diverse dataset featuring South Asian skin tones without makeup.
- Use lightweight algorithms to ensure the system works efficiently on mobile applications.
- Provide a color palette recommendation alongside the classification.

Challenges

- Lack of public datasets for South Asian seasonal color analysis
- Ensuring model accuracy with small datasets
- Addressing diverse skin tones in South Asia

Methodology

Step 1: Data Augmentation

Due to the dataset being small (25 instances per class) we had to augment the data.

```python
# Function to augment data
def augment_data(df, num_augmentations=1):
    augmented_data = []
    for _, row in df.iterrows():
        for _ in range(num_augmentations):
            new_row = row.copy()
            for col in ['Eye Color_R', 'Eye Color_G', 'Eye Color_B',
                        'Eyebrow Color_R', 'Eyebrow Color_G', 'Eyebrow Color_B',
                        'Skin Color_R', 'Skin Color_G', 'Skin Color_B',
                        'Lip color_R', 'Lip color_G', 'Lip color_B']:
                # Add small random noise to R, G, B values
                new_row[col] = min(max(new_row[col] + np.random.randint(-5, 6), 0), 255)
            augmented_data.append(new_row)
    return pd.DataFrame(augmented_data)
```

Output:

```
Original dataset size: 100
Augmented dataset size: 200
```

After augmentation, we had 50 instances per class.

Step 2: Image Preprocessing

- Detect face, eyes, and eyebrows using MTCNN
- Crop regions for face, eyes, and eyebrows
- Apply KMeans clustering to extract dominant RGB colors
- Display dominant color swatches for each region

Step 3: Features

- Skin Tone RGB value
- Eye Color RGB value
- Eyebrow Color RGB value
- Lip Color RGB value

```python
# Function to calculate brightness
def calculate_brightness(r, g, b):
    return (0.299 * r + 0.587 * g + 0.114 * b) / 255
```

- Brightness for Skin tone, Lip color, Eye color, and Eyebrow color.
- In total we had 8 features.

Step 4: Train Test Split

- We split the dataset into 80% training and 20% testing while ensuring that equal instances from each class are present in the training set.

```
Training set class distribution:
 Season
Spring    40
Winter    40
Autumn    40
Summer    40
Name: count, dtype: int64
Testing set class distribution:
 Season
Autumn    10
Summer    10
Spring    10
Winter    10
Name: count, dtype: int64
```

Step 5: Model Training and Evaluation

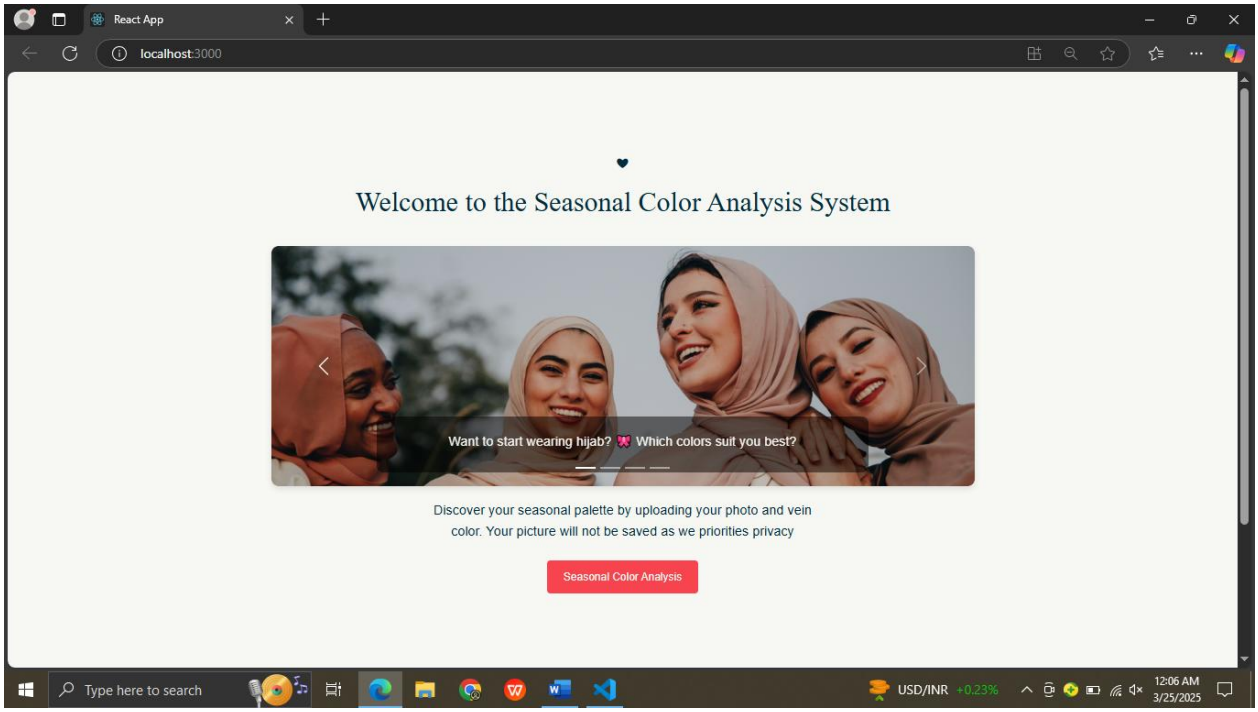
- Trained models Random Forest and got 92.5% accuracy.
- Evaluate performance using accuracy, precision, recall, and F1 score.

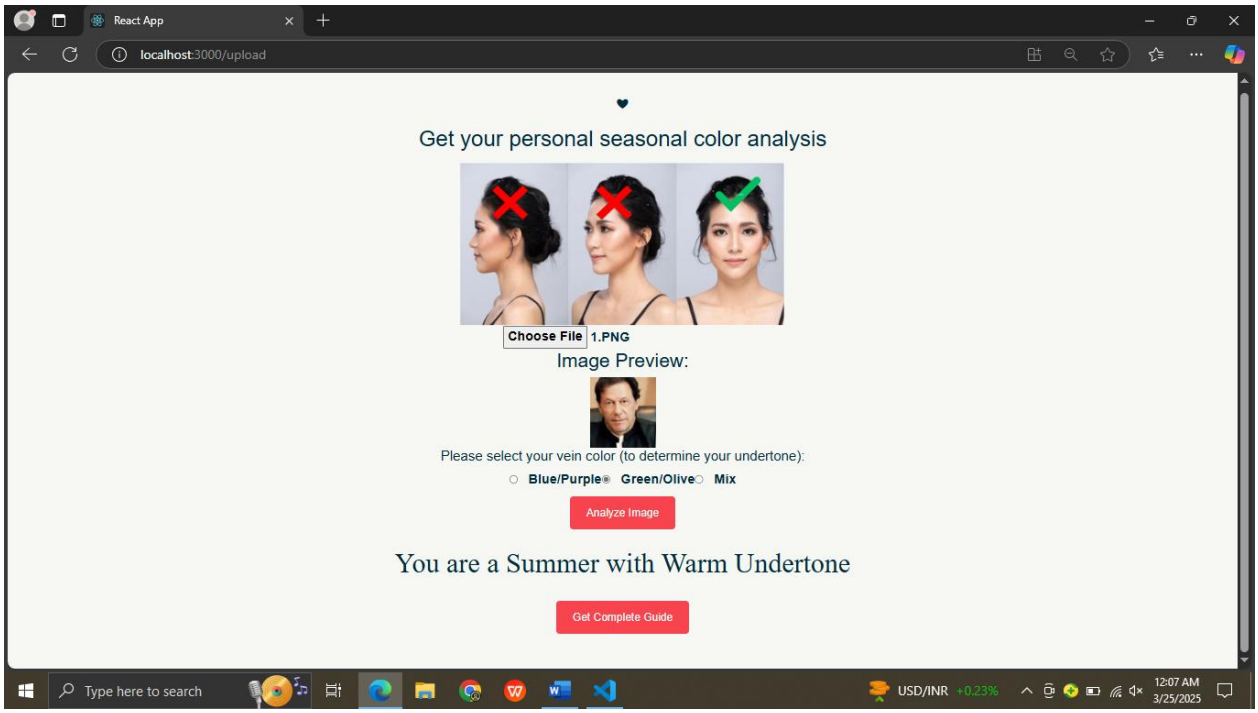
```
Random Forest Accuracy: 0.925
Random Forest Classification Report:
              precision    recall  f1-score   support

      Autumn       1.00      1.00      1.00        10
      Spring       1.00      0.90      0.95        10
      Summer       0.89      0.80      0.84        10
      Winter       0.83      1.00      0.91        10

    accuracy                           0.93        40
   macro avg       0.93      0.93      0.92        40
weighted avg       0.93      0.93      0.92        40
```

Step 6: Make the frontend and connect the Random Forest model.

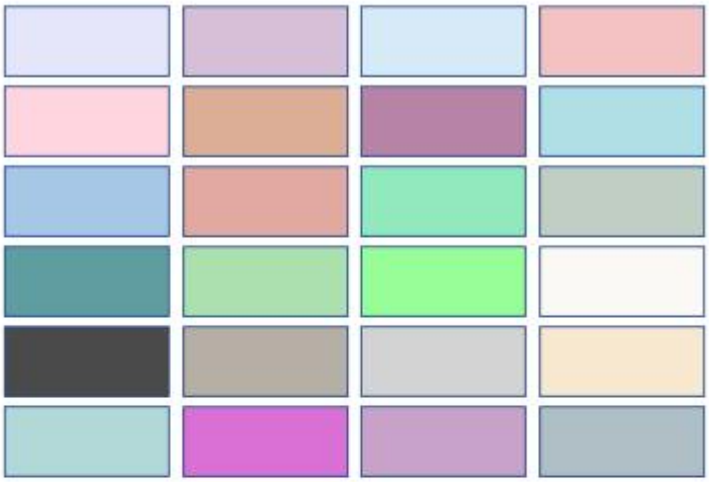
- We used React JS for frontend and connected the model via Flask

Interface:


React App
localhost:3000
Welcome to the Seasonal Color Analysis System
Want to start wearing hijab? Which colors suit you best?
Discover your seasonal palette by uploading your photo and vein color. Your picture will not be saved as we priorities privacy
Seasonal Color Analysis
Type here to search
USD/INR +0.23%
12:06 AM
3/25/2025


React App
localhost:3000/upload
Get your personal seasonal color analysis
Choose File 1.PNG
Image Preview:
Please select your vein color (to determine your undertone):
Blue/Purple
Green/Olive
Mix
Analyze Image
You are a Summer with Warm Undertone
Get Complete Guide
Type here to search
USD/INR +0.23%
12:07 AM
3/25/2025

A Guide for Summer with warm Undertone



Flattering Colors:	Less Flattering Colors:
Cool Pastels: Lavender, ice blue, soft pink. Muted Tones: Dusty rose, mauve, powder blue. Neutral Tones: Charcoal gray, soft white. Soft Greens: Seafoam green, sage green.	Bright Warm Tones: Orange, golden yellow (clash with your cool undertone). Dark Muted Tones: Burgundy, olive green (can overwhelm your soft palette). Warm Neutrals: Camel, warm beige (can look muddy).
Jewelry Recommendations: Silver: Works well with your neutral-cool leaning. White Gold: A sleek, modern choice. Rose Gold: Can work in small doses. Gold: Can work in small doses.	
Additional Tips: Experiment with both warm and cool makeup tones, but lean slightly cooler. Hair colors with neutral undertones (e.g., light brown, ash blonde) work best.	

Overall Final Working: