

## “App.py”: Main app python code

```
import os
from flask import Flask, render_template, Response, request, jsonify
import cv2
import numpy as np
from scipy.spatial import KDTree
import pyttsx3
from Utilpt2 import get_limits, get_color_bgr_values, get_color_objects, get_color_makeup

app = Flask(__name__)

# Global variables and initialization
clicked_color = None
clicked_position = (-1, -1)
color_box_size = 150
show_speaker = False
average_area_size = 1 # Scan size set to one pixel
current_frame = None
colors = get_color_bgr_values()
lab_colors = {name: cv2.cvtColor(np.uint8([[value]]), cv2.COLOR_BGR2Lab)[0][0] for name, value in colors.items()}
lab_values = np.array(list(lab_colors.values()))
kdtree = KDTree(lab_values)
objects = get_color_objects()
favorites = [] # List to store favorite objects

def speak(text):

    engine = pyttsx3.init()
    engine.setProperty('rate', 150) # Set speech rate
    engine.setProperty('volume', 1.0) # Set volume level between 0.0 and 1.0
    engine.say(text)
    engine.runAndWait()

def get_color_name(bgr_color):
    clicked_lab = cv2.cvtColor(np.uint8([[bgr_color]]), cv2.COLOR_BGR2Lab)[0][0]
    _, index = kdtree.query(clicked_lab)
    color_name = list(lab_colors.keys())[index]
    return color_name

def average_color(image, position, size):
    x, y = position
    height, width, _ = image.shape
    x_start = max(x - size, 0)
    x_end = min(x + size, width)
    y_start = max(y - size, 0)
    y_end = min(y + size, height)
    area = image[y_start:y_end, x_start:x_end]

    # Apply Gaussian blur to the selected area with higher precision
    blurred_area = cv2.GaussianBlur(area, (15, 15), 0)
```

```

# Calculate the average color in the area
average_color = cv2.mean(blurred_area)[:3]
return np.array(average_color, dtype=np.uint8)

```

```

@app.route('/')
def index():
    return render_template('index.html')

```

```

def generate_frames():
    global current_frame
    cap = cv2.VideoCapture(0)

```

```

# Set auto-focus mode to continuous
cap.set(cv2.CAP_PROP_AUTOFOCUS, 1)

```

```

while True:
    success, frame = cap.read()
    if not success:
        break
    else:
        # Brighten the frame by scaling the pixel values
        frame = cv2.convertScaleAbs(frame, alpha=1.1, beta=5) # Increase alpha and beta to brighten the image

```

```

# Remove green circle and highlight the clicked position
if clicked_position != (-1, -1):
    x, y = clicked_position
    cv2.rectangle(frame, (x - 1, y - 1), (x + 1, y + 1), (0, 0, 255), 2) # Draw a small red box

```

```

current_frame = frame.copy()
ret, buffer = cv2.imencode('.jpg', frame)
frame = buffer.tobytes()
yield (b'--frame\r\n'
        b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')

```

```

@app.route('/video_feed')
def video_feed():
    return Response(generate_frames(), mimetype='multipart/x-mixed-replace; boundary=frame')

```

```

@app.route('/click_event', methods=['POST'])
def click_event():
    data = request.json
    x = int(data['x'])
    y = int(data['y'])
    global clicked_position, clicked_color, current_frame
    clicked_position = (x, y)
    if current_frame is not None:
        clicked_color = average_color(current_frame, clicked_position, average_area_size)
        color_name = get_color_name(clicked_color)
        color_makeup = get_color_makeup().get(color_name, [])
        darkness_level = calculate_darkness(clicked_color)
        objects_list = objects.get(color_name, [])

```

```

print(f"Clicked Position: {clicked_position}, Clicked Color (BGR): {clicked_color}, Detected Color Name: {color_name}")

response = {
    "color_name": color_name,
    "color_makeup": color_makeup,
    "darkness_level": darkness_level,
    "objects_list": objects_list
}
return jsonify(response)
return "", 204

@app.route('/speak', methods=['POST'])
def speak_text():
    data = request.json
    text = data['text']
    print(f"Speaking: {text}") # Add logging for debugging
    speak(text)
    return jsonify({"status": "success"})

@app.route('/save_favorite', methods=['POST'])
def save_favorite():
    data = request.json
    object_name = data['object_name']
    color_name = data['color_name']
    favorites.append({'object_name': object_name, 'color_name': color_name})
    return jsonify({"object_name": object_name, "color_name": color_name})

@app.route('/delete_favorite', methods=['POST'])
def delete_favorite():
    data = request.json
    object_name = data['object_name']
    color_name = data['color_name']
    global favorites
    favorites = [favorite for favorite in favorites if
        not (favorite['object_name'] == object_name and favorite['color_name'] == color_name)]
    return jsonify({"success": True})

@app.route('/feedback', methods=['POST'])
def feedback():
    data = request.json
    is_accurate = data['accurate']
    # Handle feedback to improve future detections
    return jsonify({"status": "success", "accurate": is_accurate})

def calculate_darkness(bgr_color):
    gray_color = cv2.cvtColor(np.uint8([[bgr_color]]), cv2.COLOR_BGR2GRAY)[0][0]
    return (255 - gray_color) / 255 * 10

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=int(os.environ.get('PORT', 5000)))

```

# “Index.html”: Web app interface

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Clarity - Color Detection</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <style>
    @import url('https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap');

    body {
      font-family: 'Roboto', sans-serif;
      background: linear-gradient(135deg, #f06, #ff9);
      margin: 0;
      padding: 0;
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      min-height: 100vh;
      overflow-x: hidden;
      animation: gradient 5s infinite alternate;
    }

    body.color-blind-friendly {
      background: linear-gradient(135deg, #000, #fff);
      filter: grayscale(100%);
      color: #000;
    }

    header {
      display: flex;
      align-items: center;
      justify-content: center;
      background: transparent;
      width: 100%;
      padding: 20px;
    }

    header img {
      height: 100px;
    }

    #app-title {
      color: #fff;
      font-size: 2rem;
      margin-top: 20px;
      animation: fadeInUp 2s ease-out;
      text-align: center;
    }
```

```
#app-title.color-blind-friendly {
  color: #000;
  background: rgba(255, 255, 255, 0.8);
  padding: 10px 20px;
  border-radius: 10px;
}
```

```
#video-container {
  position: relative;
  border: 2px solid #333;
  border-radius: 10px;
  overflow: hidden;
  margin: 20px 0;
  animation: fadeIn 1.5s ease-in;

  width: 90%;
  max-width: 640px;
}
```

```
#video-container.color-blind-friendly {
  filter: grayscale(100%);
}
```

```
#video {
  display: block;
  width: 100%;
}
```

```
.button-container {
  margin-bottom: 20px;
  display: flex;
  gap: 10px;
  animation: fadeIn 2s ease-in;
  flex-wrap: wrap;
  justify-content: center;
}
```

```
button {
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  background-color: #333;
  color: #fff;
  cursor: pointer;
  transition: background-color 0.3s, transform 0.3s;
}
```

```
button:hover {
  background-color: #555;
  transform: scale(1.05);
}
```

```
button.color-blind-friendly {
```

```

    background-color: #000;
    color: #fff;
}

.button-on {
    background-color: #0f0 !important;
}

.button-off {
    background-color: #f00 !important;
}

#color-info {
    background-color: rgba(255, 255, 255, 0.9);
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    text-align: left;
    width: 90%;
    max-width: 300px;
    margin-bottom: 20px;
    animation: fadeInUp 1s ease-out;
}

#color-info.color-blind-friendly {
    background-color: rgba(0, 0, 0, 0.9);
    color: #fff;
}

#color-info p {
    margin: 10px 0;
}

#color-info span {
    font-weight: bold;
}

.clicked-box {
    position: absolute;
    border: 2px solid red;
    width: 10px;
    height: 10px;
    pointer-events: none;
    animation: pulse 1s infinite;
}

.clicked-box.color-blind-friendly {
    border-color: #000;
}

.scale-container {
    margin-top: 10px;
}

```

```
width: 100%;
}

.scale-bar {
width: 100%;
height: 10px;
background-color: #ddd;
position: relative;
border-radius: 5px;
overflow: hidden;
}

.scale-bar.color-blind-friendly {
background-color: #000;
}

.scale-marker {
height: 20px;
width: 2px;
background-color: #333;
position: absolute;
top: -5px;
transition: left 0.5s;
}

.scale-marker.color-blind-friendly {
background-color: #fff;
}

.scale-labels {
display: flex;
justify-content: space-between;
margin-top: 5px;
}

.scale-label {
position: relative;
width: 10%;
text-align: center;
}

.instructions {
background-color: rgba(255, 255, 255, 0.9);
padding: 20px;
border-radius: 10px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
width: 90%;
max-width: 300px;
margin-top: 20px;
text-align: left;
}

.instructions.color-blind-friendly {
```

```
background-color: rgba(0, 0, 0, 0.9);
color: #fff;
}
```

```
.favorites-container {
background-color: rgba(255, 255, 255, 0.9);
padding: 20px;
border-radius: 10px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
width: 90%;
max-width: 300px;
margin-top: 20px;
text-align: left;
}
```

```
.favorites-container.color-blind-friendly {
background-color: rgba(0, 0, 0, 0.9);
color: #fff;
}
```

```
.favorites-container ul {
list-style-type: none;
padding: 0;
}
```

```
.favorites-container li {
margin: 5px 0;
display: flex;
justify-content: space-between;
align-items: center;
}
```

```
.delete-button {
background-color: red;
border: none;
color: white;
padding: 5px;
cursor: pointer;
border-radius: 5px;
}
```

```
.delete-button.color-blind-friendly {
background-color: #000;
}
```

```
@keyframes fadeIn {
from {
opacity: 0;
}
to {
opacity: 1;
}
}
```



```

@keyframes fadeInUp {
  from {
    transform: translateY(20px);
    opacity: 0;
  }
  to {
    transform: translateY(0);
    opacity: 1;
  }
}

@keyframes pulse {
  0% {
    transform: scale(1);
  }
  50% {
    transform: scale(1.2);
  }
  100% {
    transform: scale(1);
  }
}
</style>
</head>
<body>
  <header>
    
  </header>
  <div id="app-title" class="color-blind-friendly">
    Color Detection App
  </div>
  <div id="video-container">
    
    <div id="click-box" class="clicked-box" style="display: none;"></div>
  </div>
  <div class="button-container">
    <button id="speak-button"><img alt="Speaker icon" data-bbox="295 638 315 655"/> Speak</button>
    <button id="toggle-color-blind-mode">Toggle Color-Blind Mode</button>
  </div>
  <div id="color-info">
    <p>Color: <span id="color-name"></span></p>
    <p>Makeup: <span id="color-makeup"></span></p>
    <p>Darkness Level: <span id="darkness-level"></span></p>
    <div class="scale-container">
      <div class="scale-bar">
        <div class="scale-marker" id="scale-marker"></div>
      </div>
      <div class="scale-labels" id="scale-labels">
        <!-- Dynamic labels will be added here -->
      </div>
    </div>
    <p>Associated Objects: <span id="objects-list"></span></p>
  </div>
  <form id="favorite-form" class="favorites-container">

```

```

<h2>Favorite Objects</h2>
<input type="text" id="object-name" placeholder="Enter object name" required>
<button type="submit">Save</button>
<ul id="favorites-list"></ul>
</form>
<div class="instructions">
  <h2>Instructions</h2>
  <p>1. Click on the video feed to select a color.</p>
  <p>2. The color information and associated objects will be displayed.</p>
  <p>3. Use the "Speak" button to hear the color description.</p>
  <p>4. Enter an object name and click "Save" to save the color information.</p>
</div>
<script>
  const video = document.getElementById('video');
  const clickBox = document.getElementById('click-box');
  const scaleMarker = document.getElementById('scale-marker');
  const scaleLabels = document.getElementById('scale-labels');
  const favoriteForm = document.getElementById('favorite-form');
  const favoritesList = document.getElementById('favorites-list');
  const toggleColorBlindMode = document.getElementById('toggle-color-blind-mode');
  const speakButton = document.getElementById('speak-button');
  let selectedColorInfo = ""; // Initialize empty selected color info

  video.onclick = function(event) {
    const rect = video.getBoundingClientRect();
    const x = Math.round(event.clientX - rect.left);
    const y = Math.round(event.clientY - rect.top);

    // Position the click box
    clickBox.style.left = `${x - 5}px`;
    clickBox.style.top = `${y - 5}px`;
    clickBox.style.display = 'block';
    console.log(`Clicked at (${x}, ${y})`);

    // Reset the selectedColorInfo to empty when a new color is scanned
    selectedColorInfo = "";

    fetch('/click_event', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({ x: x, y: y })
    })
    .then(response => response.json())
    .then(data => {
      document.getElementById('color-name').innerText = data.color_name;
      document.getElementById('color-makeup').innerText = data.color_makeup.join(', ');
      document.getElementById('darkness-level').innerText = data.darkness_level.toFixed(1);
      document.getElementById('objects-list').innerText = data.objects_list.join(', ');

      // Save the selected color information for speaking
      selectedColorInfo = `Color name: ${data.color_name}. Makeup: ${data.color_makeup.join(', ')}. Darkness Level:
      ${data.darkness_level.toFixed(1)}. Associated Objects: ${data.objects_list.join(', ')}`;
    });
  };

```

```

// Update the scale marker based on darkness level
scaleMarker.style.left = `${data.darkness_level * 10}%`;
console.log(`Darkness Level: ${data.darkness_level}`);
})
.catch(error => console.error('Error:', error));
};

// Function to speak the selected color information
speakButton.onclick = function() {
  if (selectedColorInfo) {
    console.log(`Attempting to speak: ${selectedColorInfo}`); // Log the attempt to speak
    fetch('/speak', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({ text: selectedColorInfo })
    })
    .then(response => {
      if (response.ok) {
        console.log("Successfully sent text to speak.");
      } else {
        console.error("Failed to send text to speak.");
      }
      return response.json(); // Ensure response is returned and processed
    })
    .then(data => console.log('Speak status:', data))
    .catch(error => console.error('Error:', error));
  }
};

```

```

favoriteForm.onsubmit = function(event) {
  event.preventDefault();
  const objectName = document.getElementById('object-name').value;
  const colorName = document.getElementById('color-name').innerText;
  if (objectName && colorName) {
    fetch('/save_favorite', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({ object_name: objectName, color_name: colorName })
    })
    .then(response => response.json())
    .then(data => {
      const li = document.createElement('li');
      li.textContent = `${data.object_name} - ${data.color_name}`;
      const deleteButton = document.createElement('button');
      deleteButton.textContent = 'Delete';
      deleteButton.className = 'delete-button';
      deleteButton.onclick = function() {
        fetch('/delete_favorite', {
          method: 'POST',
          headers: {

```

```

        'Content-Type': 'application/json'
    },
    body: JSON.stringify({ object_name: data.object_name, color_name: data.color_name })
  })
  .then(response => response.json())
  .then(result => {
    if (result.success) {
      favoritesList.removeChild(li);
    }
  })
  .catch(error => console.error('Error:', error));
};
li.appendChild(deleteButton);
favoritesList.appendChild(li);
document.getElementById('object-name').value = ""; // Clear the input
})
.catch(error => console.error('Error:', error));
}
};

toggleColorBlindMode.onclick = function() {
  document.body.classList.toggle('color-blind-friendly');
  document.getElementById('app-title').classList.toggle('color-blind-friendly');
  document.getElementById('video-container').classList.toggle('color-blind-friendly');
  document.getElementById('color-info').classList.toggle('color-blind-friendly');
  document.getElementById('click-box').classList.toggle('color-blind-friendly');
  document.querySelector('.scale-bar').classList.toggle('color-blind-friendly');
  document.querySelector('.scale-marker').classList.toggle('color-blind-friendly');
  document.querySelector('.instructions').classList.toggle('color-blind-friendly');
  document.querySelector('.favorites-container').classList.toggle('color-blind-friendly');
  document.querySelectorAll('.delete-button').forEach(button => button.classList.toggle('color-blind-friendly'));
};

// Create scale labels dynamically
function createScaleLabels() {
  for (let i = 1; i <= 10; i++) {
    const label = document.createElement('div');
    label.className = 'scale-label';
    label.textContent = i;
    scaleLabels.appendChild(label);
  }
}

createScaleLabels();
</script>
</body>
</html>

```

## “Utilpt2.py”: All the color names with their composition

```

import numpy as np

import cv2

```

```

def get_limits(color):
    # Convert RGB to BGR
    color_bgr = color[::-1] # Reverse the order from RGB to BGR

    c = np.uint8([color_bgr]) # Insert BGR values that you want to convert to HSV
    hsvC = cv2.cvtColor(c, cv2.COLOR_BGR2HSV)

    hue = hsvC[0][0][0]

    lowerLimit = (max(hue - 10, 0), 100, 100)
    upperLimit = (min(hue + 10, 179), 255, 255)

    lowerLimit = np.array(lowerLimit, dtype=np.uint8)
    upperLimit = np.array(upperLimit, dtype=np.uint8)

    return lowerLimit, upperLimit

def get_color_bgr_values():
    return {
        'Red': [0, 0, 255],
        'Green': [0, 255, 0],
        'Blue': [255, 0, 0],
        'Yellow': [0, 255, 255],
        'Cyan': [255, 255, 0],
        'Magenta': [255, 0, 255],
        'White': [255, 255, 255],
        'Black': [0, 0, 0],
        'Gray': [128, 128, 128],
        'Orange': [0, 165, 255],
        'Pink': [203, 192, 255],
        'Purple': [128, 0, 128],
        'Brown': [42, 42, 165],
        'Maroon': [0, 0, 128],
        'Olive': [0, 128, 128],
        'Lime': [0, 255, 0],
        'Teal': [128, 128, 0],
        'Navy': [128, 0, 0],
        'Aquamarine': [212, 255, 127],
        'Azure': [240, 255, 255],
        'Beige': [220, 245, 245],
        'Bisque': [196, 228, 255],
        'Chocolate': [30, 105, 210],
        'Coral': [80, 127, 255],
        'Crimson': [60, 20, 220],
        'Fuchsia': [255, 0, 255],
        'Gold': [0, 215, 255],
        'Ivory': [240, 255, 255],
        'Khaki': [140, 230, 240],
        'Lavender': [230, 230, 250],
        'Linen': [250, 240, 230],
        'Moccasin': [181, 228, 255],
        'Orchid': [214, 112, 218],
        'Peru': [63, 133, 205],
        'Plum': [221, 160, 221],
    }

```

'Salmon': [114, 128, 250],  
'Sienna': [45, 82, 160],  
'Silver': [192, 192, 192],  
'Tan': [210, 180, 140],  
'Thistle': [216, 191, 216],  
'Tomato': [71, 99, 255],  
'Turquoise': [208, 224, 64],  
'Violet': [238, 130, 238],  
'Wheat': [179, 222, 245],  
'Sky Blue': [235, 206, 135],  
'Forest Green': [34, 139, 34],  
'Indian Red': [92, 92, 205],  
'Steel Blue': [180, 130, 70],  
'Dark Slate Blue': [139, 61, 72],  
'Medium Spring Green': [0, 250, 154],  
'Rosy Brown': [188, 143, 143],  
'Dark Olive Green': [85, 107, 47],  
'Saddle Brown': [19, 69, 139],  
'Medium Purple': [147, 112, 219],  
'Medium Sea Green': [60, 179, 113],  
'Fire Brick': [34, 34, 178],  
'Cadet Blue': [160, 158, 95],  
'Pale Violet Red': [147, 112, 219],  
'Dark Khaki': [107, 183, 189],  
'Light Coral': [128, 128, 240],  
'Dark Salmon': [122, 150, 233],  
'Medium Aquamarine': [102, 205, 170],  
'Dark Orchid': [153, 50, 204],  
'Medium Orchid': [186, 85, 211],  
'Dark Turquoise': [0, 206, 209],  
'Medium Turquoise': [72, 209, 204],  
'Light Salmon': [122, 160, 255],  
'Dark Goldenrod': [11, 134, 184],  
'Medium Violet Red': [133, 21, 199],  
'Light Green': [144, 238, 144],  
'Light Sky Blue': [250, 206, 135],  
'Light Steel Blue': [222, 196, 176],  
'Pale Goldenrod': [170, 232, 238],  
'Light Yellow': [224, 255, 255],  
'Mint Cream': [250, 255, 250],  
'Honeydew': [240, 255, 240],  
'Alice Blue': [240, 248, 255],  
'Ghost White': [240, 255, 255],  
'Snow': [250, 250, 255],  
'Seashell': [255, 245, 238],  
'Floral White': [255, 250, 240],  
'Old Lace': [253, 245, 230],  
'Antique White': [250, 235, 215],  
'Papaya Whip': [255, 239, 213],  
'Blanched Almond': [255, 235, 205],  
'Lavender Blush': [255, 240, 245],  
'Cornsilk': [255, 248, 220],  
'Lemon Chiffon': [255, 250, 205],  
'Light Goldenrod Yellow': [250, 250, 210],  
'Light Cyan': [224, 255, 255],  
'Pale Green': [152, 251, 152],  
'Spring Green': [0, 255, 127],

'Sea Green': [46, 139, 87],  
'Dark Green': [0, 100, 0],  
'Yellow Green': [50, 205, 154],  
'Olive Drab': [35, 142, 107],  
'Dark Sea Green': [143, 188, 143],  
'Light Sea Green': [170, 178, 32],  
'Light Goldenrod': [85, 236, 250],  
'Goldenrod': [32, 165, 218],  
'Burlywood': [222, 184, 135],  
'Navajo White': [255, 222, 173],  
'Peach Puff': [255, 218, 185],  
'Light Pink': [255, 182, 193],  
'Deep Pink': [255, 20, 147],  
'Hot Pink': [255, 105, 180],  
'Violet Red': [255, 35, 85],  
'Medium Slate Blue': [123, 104, 238],  
'Cornflower Blue': [100, 149, 237],  
'Royal Blue': [65, 105, 225],  
'Dodger Blue': [30, 144, 255],  
'Deep Sky Blue': [0, 191, 255],  
'Light Blue': [173, 216, 230],  
'Powder Blue': [176, 224, 230],  
'Pale Turquoise': [175, 238, 238],  
'Charcoal': [54, 69, 79],  
'Dark Gray': [169, 169, 169],  
'Light Gray': [211, 211, 211],  
'Ash Gray': [178, 190, 181],  
'Jet': [52, 52, 52],  
'Smoky Black': [8, 12, 16],  
'Outer Space': [76, 74, 65],  
'Timberwolf': [210, 215, 219],  
'Platinum': [226, 228, 229],  
'Old Silver': [132, 132, 130],  
'Khaki Drab': [51, 60, 69],  
'Desert Sand': [44, 93, 102],  
'Pale Silver': [187, 192, 201],  
'Shadow': [93, 121, 138],  
'Pewter Blue': [183, 168, 139],  
'Cool Black': [99, 46, 0],  
'Cerulean': [208, 189, 163],  
'Ultramarine': [143, 10, 18],  
'Electric Blue': [0, 255, 204],  
'Jade': [107, 168, 0],  
'Persian Blue': [204, 48, 54],  
'Cerulean Blue': [205, 92, 92],  
'Egyptian Blue': [166, 52, 16],  
'Klein Blue': [255, 48, 30],  
'Space Cadet': [102, 57, 176],  
'Midnight': [25, 25, 112],  
'Gun metal': [42, 52, 57],  
'Black Olive': [59, 60, 54],  
'Davys Grey': [85, 85, 85],  
'Raisin Black': [36, 33, 36],  
'Eerie Black': [27, 27, 27],  
'Taupe': [72, 60, 50],  
'Coal': [54, 54, 54],  
'Licorice': [26, 17, 16],

'Rich Black': [1, 11, 19],  
'Vantablack': [0, 0, 0],  
'Iridium': [55, 55, 55],  
'Obsidian': [21, 22, 31],  
'Apple Green': [0, 182, 141],  
'Apricot': [177, 206, 251],  
'Aqua': [255, 255, 0],  
'Army Green': [32, 83, 75],  
'Asparagus': [107, 169, 135],  
'Atomic Tangerine': [102, 153, 255],  
'Baker-Miller Pink': [175, 145, 255],  
'Banana Yellow': [53, 225, 255],  
'Barn Red': [2, 10, 124],  
'Battleship Grey': [130, 132, 132],  
'Bazaar': [123, 119, 152],  
'Blizzard Blue': [238, 229, 172],  
'Blue Bell': [208, 162, 162],  
'Blue Green': [186, 152, 13],  
'Blue Violet': [226, 43, 138],  
'Blush': [131, 93, 222],  
'Bole': [59, 68, 121],  
'Brick Red': [84, 65, 203],  
'Bright Green': [0, 255, 102],  
'Bright Lilac': [239, 145, 216],  
'Bright Maroon': [72, 33, 195],  
'Bright Navy Blue': [210, 116, 25],  
'Bright Pink': [127, 0, 255],  
'Bright Turquoise': [222, 232, 8],  
'Bright Ube': [232, 159, 209],  
'Bright Yellow': [0, 255, 255],  
'Brilliant Rose': [163, 85, 255],  
'Brink Pink': [127, 96, 251],  
'British Racing Green': [37, 66, 0],  
'Bronze': [50, 127, 205],  
'Brown Sugar': [77, 110, 175],  
'Bud Green': [97, 182, 123],  
'Buff': [130, 220, 240],  
'Burgundy': [32, 0, 128],  
'Byzantine': [164, 51, 189],  
'Byzantium': [99, 41, 112],  
'Cambridge Blue': [173, 193, 163],  
'Camel': [107, 154, 193],  
'Capri': [255, 191, 0],  
'Cardinal': [58, 30, 196],  
'Caribbean Green': [153, 204, 0],  
'Carmine': [24, 0, 150],  
'Carnation Pink': [201, 166, 255],  
'Carnelian': [27, 27, 179],  
'Carrot Orange': [33, 145, 237],  
'Ceil': [207, 161, 146],  
'Celadon': [175, 225, 172],  
'Celeste': [255, 255, 178],  
'Cerise': [99, 49, 222],  
'Cerulean Frost': [195, 155, 109],  
'Champagne': [206, 231, 247],  
'Cherry': [99, 49, 222],  
'Chestnut': [53, 69, 149],



'China Pink': [161, 111, 222],  
'Chinese Red': [30, 56, 170],  
'Chocolate (Traditional)': [0, 63, 123],  
'Cinereous': [123, 129, 152],  
'Cinnabar': [52, 66, 227],  
'Cinnamon': [30, 105, 210],  
'Citrine': [10, 208, 228],  
'Classic Rose': [231, 204, 251],  
'Cobalt Blue': [171, 71, 0],  
'Columbia Blue': [255, 221, 155],  
'Congo Pink': [121, 131, 248],  
'Copper': [51, 115, 184],  
'Copper Rose': [102, 102, 153],  
'Coquelicot': [0, 56, 255],  
'Coral Pink': [121, 131, 248],  
'Cordovan': [69, 63, 137],  
'Cornflower': [237, 149, 100],  
'Cosmic Cobalt': [136, 45, 46],  
'Cosmic Latte': [231, 248, 255],  
'Cotton Candy': [217, 188, 255],  
'Cream': [208, 253, 255],  
'Crimson Red': [0, 0, 153],  
'Cyan Blue': [235, 183, 0],  
'Cyber Grape': [124, 66, 88],  
'Cyber Yellow': [0, 211, 255],  
'Daffodil': [49, 255, 255],  
'Dandelion': [48, 225, 240],  
'Dark Brown': [33, 67, 101],  
'Dark Candy Apple Red': [0, 0, 164],  
'Dark Cerulean': [126, 69, 8],  
'Dark Chestnut': [96, 105, 152],  
'Dark Coral': [69, 91, 205],  
'Dark Cyan': [139, 139, 0],  
'Dark Electric Blue': [120, 104, 83],  
'Dark Imperial Blue': [106, 65, 0],  
'Dark Jungle Green': [33, 36, 26],  
'Dark Lava': [50, 60, 72],  
'Dark Lavender': [150, 79, 115],  
'Dark Liver': [79, 75, 83],  
'Dark Magenta': [139, 0, 139],  
'Dark Medium Gray': [169, 169, 169],  
'Dark Midnight Blue': [102, 51, 0],  
'Dark Moss Green': [35, 93, 74],  
'Dark Pastel Green': [60, 192, 3],  
'Dark Red': [0, 0, 139],  
'Dark Scarlet': [25, 3, 86],  
'Dark Sienna': [20, 20, 60],  
'Dark Spring Green': [69, 114, 23],  
'Dark Tan': [81, 129, 145],  
'Dark Tangerine': [18, 168, 255],  
'Dark Terracotta': [92, 78, 204],  
'Dark Violet': [211, 0, 148],  
'Jet Black': [0, 0, 0],  
'Ebony': [55, 71, 79],  
'Onyx': [15, 15, 15],  
'Obsidian': [21, 22, 31],  
'Dark Lava': [72, 60, 50],

```
}
```

```
def get_color_objects():  
    return {  
        'Red': ['Apple', 'Tomato', 'Firetruck'],  
        'Green': ['Leaf', 'Frog', 'Grass'],  
        'Blue': ['Sky', 'Ocean', 'Blueberries'],  
        'Yellow': ['Banana', 'Sunflower', 'Lemon'],  
        'Cyan': ['Tropical Waters', 'Blue Jay Feathers', 'Swimming Pool'],  
        'Magenta': ['Raspberries', 'Orchid', 'Beet Juice'],  
        'White': ['Snow', 'Cloud', 'Cotton'],  
        'Black': ['Coal', 'Obsidian', 'Night'],  
        'Gray': ['Stone', 'Ash'],  
        'Orange': ['Orange', 'Carrot', 'Pumpkin'],  
        'Pink': ['Flamingo', 'Cotton Candy', 'Rose'],  
        'Purple': ['Eggplant', 'Grapes', 'Plums'],  
        'Brown': ['Chocolate', 'Wood', 'Coffee'],  
        'Maroon': ['Red wine', 'Leather', 'Pomegranate'],  
        'Olive': ['Olives', 'Moss'],  
        'Lime': ['Limes', 'Green Apples'],  
        'Teal': ['Kingfisher Bird'],  
        'Navy': ['Blue Pen Ink', 'Midnight Sky'],  
        'Aquamarine': ['Sea Glass', 'Ocean Water', 'Gemstones'],  
        'Azure': ['Clear Sky', 'Peacock', 'Blue Jays'],  
        'Beige': ['Sand', 'Khaki'],  
        'Bisque': ['Porcelain Dolls', 'Eggshells'],  
        'Chocolate': ['Milk Chocolate', 'Chocolate Cake'],  
        'Coral': ['Coral Reef', 'Tropical Fish'],  
        'Crimson': ['Dark Rose', 'Blood'],  
        'Fuchsia': ['Dragon fruit', 'Butterflies'],  
        'Gold': ['Gold Nugget', 'Coins', 'Rings'],  
        'Ivory': ['Elephant Tusk', 'Animal Teeth'],  
        'Khaki': ['Sand', 'Mushrooms', 'Camel Fur'],  
        'Lavender': ['Lavender Flower', 'Sunsets'],  
        'Linen': ['Cloth', 'Bedsheets', 'Napkins'],  
        'Moccasin': ['Fur Slippers'],  
        'Orchid': ['Orchids'],  
        'Peru': ['Autumn Leaves', 'Rust', 'Brown Sugar'],  
        'Plum': ['Plums', 'Eggplant', 'Beetroot'],  
        'Salmon': ['Salmon', 'Sunsets', 'Flamingo'],  
        'Sienna': ['Bricks', 'Dried Leaves'],  
        'Silver': ['Quarter', 'Nickel'],  
        'Tan': ['Sand', 'Leather', 'Birch'],  
        'Thistle': ['Thistle Flower'],  
        'Tomato': ['Tomato'],  
        'Turquoise': ['Gem', 'Parrots'],  
        'Violet': ['Violet'],  
        'Wheat': ['Wheat'],  
        'Sky Blue': ['Sky'],  
        'Forest Green': ['Pine Tree'],  
        'Indian Red': ['Indian Sunset'],  
        'Steel Blue': ['Blue Heron Feathers', 'Blueberries'],  
        'Dark Slate Blue': ['Slate Stone'],
```

'Medium Spring Green': ['Spring Grass'],  
'Rosy Brown': ['Rosy Cheeks'],  
'Dark Olive Green': ['Olive Tree', 'Olives'],  
'Saddle Brown': ['Horse Fur', 'Chestnuts'],  
'Medium Purple': ['Purple Iris'],  
'Medium Sea Green': ['Seaweed'],  
'Fire Brick': ['Brick Wall'],  
'Cadet Blue': ['Blue Jay Feathers', 'Gray Whale Skin'],  
'Pale Violet Red': ['Violet Rose'],  
'Dark Khaki': ['Khaki Jacket'],  
'Light Coral': ['Coral Sand'],  
'Dark Salmon': ['Salmon Fillet'],  
'Medium Aquamarine': ['Aquamarine Gem'],  
'Dark Orchid': ['Orchid Flower'],  
'Medium Orchid': ['Orchid Flower'],  
'Dark Turquoise': ['Turquoise Gem'],  
'Medium Turquoise': ['Turquoise Sea'],  
'Light Salmon': ['Salmon Roe'],  
'Dark Goldenrod': ['Goldenrod Flower'],  
'Medium Violet Red': ['Violet Petal'],  
'Light Green': ['Green Apple'],  
'Light Sky Blue': ['Sky'],  
'Light Steel Blue': ['Steel Metal'],  
'Pale Goldenrod': ['Goldenrod Flower'],  
'Light Yellow': ['Buttercup'],  
'Mint Cream': ['Mint Leaf'],  
'Honeydew': ['Honeydew Melon'],  
'Alice Blue': ['Bluebell'],  
'Ghost White': ['Ghost Orchid'],  
'Snow': ['Snowflake'],  
'Seashell': ['Seashell'],  
'Floral White': ['White Rose'],  
'Old Lace': ['Lace Dress'],  
'Antique White': ['Antique Vase'],  
'Papaya Whip': ['Papaya Fruit'],  
'Blanched Almond': ['Almond'],  
'Lavender Blush': ['Lavender Flower'],  
'Cornsilk': ['Corn Husk'],  
'Lemon Chiffon': ['Lemon Pie'],  
'Light Goldenrod Yellow': ['Goldenrod Flower'],  
'Light Cyan': ['Cyan Gem'],  
'Pale Green': ['Green Apple'],  
'Spring Green': ['Spring Leaves'],  
'Sea Green': ['Seaweed'],  
'Dark Green': ['Evergreen'],  
'Yellow Green': ['Green Banana'],  
'Olive Drab': ['Olive Tree'],  
'Dark Sea Green': ['Sea Turtle'],  
'Light Sea Green': ['Sea Glass'],  
'Light Goldenrod': ['Goldenrod Flower'],  
'Goldenrod': ['Goldenrod Flower'],  
'Rosy Brown': ['Rosy Cheeks'],  
'Burlywood': ['Burlap'],  
'Navajo White': ['Navajo Blanket'],  
'Peach Puff': ['Peach Fruit'],  
'Light Pink': ['Pink Rose'],  
'Deep Pink': ['Pink Hibiscus'],

'Hot Pink': ['Pink Flamingo'],  
'Violet Red': ['Violet Flower'],  
'Medium Slate Blue': ['Slate Stone'],  
'Cornflower Blue': ['Cornflower'],  
'Royal Blue': ['Royal Ribbon'],  
'Dodger Blue': ['Dodger Stadium'],  
'Deep Sky Blue': ['Sky'],  
'Light Blue': ['Bluebell'],  
'Powder Blue': ['Powder Puff'],  
'Pale Turquoise': ['Turquoise Gem'],  
'Sky Blue': ['Sky'],  
'Pale Goldenrod': ['Goldenrod Flower'],  
'Peach Puff': ['Peach Fruit'],  
'Pale Green': ['Green Apple'],  
'Pale Turquoise': ['Turquoise Gem'],  
'Charcoal': ['Charcoal Briquette'],  
'Dark Gray': ['Gray Cloud'],  
'Light Gray': ['Gray Cloud'],  
'Ash Gray': ['Ash Tree'],  
'Jet': ['Jet Plane'],  
'Smoky Black': ['Smoky Quartz'],  
'Outer Space': ['Outer Space'],  
'Gun Metal': ['Gun Barrel'],  
'Timberwolf': ['Timberwolf Fur'],  
'Platinum': ['Platinum Ring'],  
'Old Silver': ['Silver Coin'],  
'Khaki Drab': ['Khaki Pants'],  
'Desert Sand': ['Desert Dune'],  
'Pale Silver': ['Silver Coin'],  
'Shadow': ['Shadow'],  
'Pewter Blue': ['Pewter Mug'],  
'Cool Black': ['Black Ice'],  
'Cerulean': ['Cerulean Sky'],  
'Ultramarine': ['Ultramarine Sea'],  
'Electric Blue': ['Blue Neon Light'],  
'Jade': ['Jade Stone'],  
'Persian Blue': ['Persian Rug'],  
'Cerulean Blue': ['Cerulean Sky'],  
'Egyptian Blue': ['Egyptian Painting'],  
'Klein Blue': ['Klein Bottle'],  
'Space Cadet': ['Cadet Uniform'],  
'Charcoal': ['Charcoal Briquette'],  
'Dark Gray': ['Gray Cloud'],  
'Light Gray': ['Gray Cloud'],  
'Outer Space': ['Outer Space'],  
'Old Silver': ['Coin'],  
'Khaki Drab': ['Dress Pants'],  
'Desert Sand': ['Desert Dune'],  
'Rosy Brown': ['Rosy Cheeks'],  
'Shadow': ['Shadow'],  
'Saddle Brown': ['Saddle Leather'],  
'Chocolate': ['Chocolate Bar'],  
'Peru': ['Rich Soil', 'Certain Types of Rock'],  
'Apple Green': ['Apple', 'Green Bell Pepper'],  
'Apricot': ['Apricot', 'Peach'],  
'Aqua': ['Swimming Pool', 'Tropical Sea'],  
'Army Green': ['Army Uniform', 'Camouflage Gear'],

'Asparagus': ['Asparagus', 'Broccoli'],  
'Atomic Tangerine': ['Tangerine', 'Orange Soda'],  
'Baker-Miller Pink': ['Bubblegum', 'Cotton Candy'],  
'Banana Yellow': ['Banana', 'Yellow Balloon'],  
'Barn Red': ['Barn', 'Old Brick House'],  
'Battleship Grey': ['Battleship', 'Military Equipment'],  
'Bazaar': ['Vintage Sofa', 'Old Photograph'],  
'Blizzard Blue': ['Winter Jacket', 'Ice Sculpture'],  
'Blue Bell': ['Bluebell Flower', 'Lavender Plant'],  
'Blue Green': ['Peacock Feather', 'Turquoise Jewelry'],  
'Blue Violet': ['Violet Flower', 'Amethyst Crystal'],  
'Blush': ['Blush Makeup', 'Rose Petal'],  
'Bole': ['Tree Bark', 'Clay Pot'],  
'Brick Red': ['Brick Wall', 'Clay Roof Tile'],  
'Bright Green': ['Neon Sign', 'Highlighter'],  
'Bright Lilac': ['Lilac Flower', 'Easter Egg'],  
'Bright Maroon': ['Maroon Sweater', 'Plum'],  
'Bright Navy Blue': ['Sailor Uniform', 'Business Suit'],  
'Bright Pink': ['Flamingo', 'Pink Neon Sign'],  
'Bright Turquoise': ['Turquoise Necklace', 'Beach Towel'],  
'Bright Ube': ['Ube Halaya', 'Purple Yam'],  
'Bright Yellow': ['Sunflower', 'Yellow Raincoat'],  
'Brilliant Rose': ['Rose Flower', 'Lipstick'],  
'Brink Pink': ['Pink Crayon', 'Carnation'],  
'British Racing Green': ['Race Car', 'Hunter Boots'],  
'Bronze': ['Bronze Statue', 'Bronze Medal'],  
'Brown Sugar': ['Brown Sugar', 'Cinnamon Stick'],  
'Bud Green': ['Leaf Bud', 'Green Bean'],  
'Buff': ['Buff Paper', 'Beige Sweater'],  
'Burgundy': ['Wine', 'Burgundy Velvet Dress'],  
'Byzantine': ['Byzantine Mosaic', 'Purple Robe'],  
'Byzantium': ['Byzantium Tapestry', 'Royal Garment'],  
'Cambridge Blue': ['University Scarf', 'Vintage Book'],  
'Camel': ['Camel Coat', 'Sand Dune'],  
'Capri': ['Capri Pants', 'Ocean Wave'],  
'Cardinal': ['Cardinal Bird', 'Red Vestment'],  
'Caribbean Green': ['Tropical Sea', 'Green Parrot'],  
'Carmine': ['Carmine Ink', 'Cherry'],  
'Carnation Pink': ['Carnation Flower', 'Birthday Cake Frosting'],  
'Carnelian': ['Carnelian Gemstone', 'Pomegranate'],  
'Carrot Orange': ['Carrot', 'Orange Juice'],  
'Ceil': ['Ceiling Paint', 'Soft Sky'],  
'Celadon': ['Celadon Pottery', 'Pistachio'],  
'Celeste': ['Celeste Cloud', 'Sky Blue Dress'],  
'Cerise': ['Cerise Ribbon', 'Cherry Blossom'],  
'Cerulean Frost': ['Winter Sky', 'Frosted Window'],  
'Champagne': ['Champagne', 'Champagne Dress'],  
'Charcoal': ['Charcoal Drawing', 'BBQ Charcoal'],  
'Cherry': ['Cherry', 'Cherry Lip Gloss'],  
'Chestnut': ['Chestnut', 'Wooden Furniture'],  
'China Pink': ['China Plate', 'Cherry Blossom'],  
'Chinese Red': ['Chinese Lantern', 'Red Envelope'],  
'Chocolate': ['Chocolate Bar', 'Brownie'],  
'Cinereous': ['Old Photograph', 'Vintage Dress'],  
'Cinnabar': ['Cinnabar Gemstone', 'Red Ink'],  
'Cinnamon': ['Cinnamon Stick', 'Cinnamon Roll'],  
'Citrine': ['Citrine Crystal', 'Lemon'],

'Classic Rose': ['Classic Rose Flower', 'Blush'],  
'Cobalt Blue': ['Cobalt Glass', 'Blue Ink'],  
'Columbia Blue': ['Columbia University T-Shirt', 'Light Blue Sweater'],  
'Congo Pink': ['Flamingo', 'Pink Bubblegum'],  
'Copper': ['Copper Wire', 'Penny'],  
'Copper Rose': ['Copper Rose', 'Metallic Paint'],  
'Coquelicot': ['Poppy Flower', 'Red Paint'],  
'Coral Pink': ['Coral Reef', 'Pink Sand'],  
'Cordovan': ['Horse Fur', 'Leather Wallet'],  
'Cornflower': ['Cornflower Flower', 'Blue Sky'],  
'Cosmic Cobalt': ['Galaxy Print', 'Night Sky'],  
'Cosmic Latte': ['Milky Way', 'Latte Foam'],  
'Cotton Candy': ['Cotton Candy', 'Pink Cloud'],  
'Cream': ['Cream', 'Vanilla Ice Cream'],  
'Crimson Red': ['Blood', 'Crimson Fabric'],  
'Cyan Blue': ['Tropical Water', 'Cyan Light'],  
'Cyber Grape': ['Grape Soda', 'Purple LED'],  
'Cyber Yellow': ['Yellow LED', 'Construction Vest'],  
'Daffodil': ['Daffodil Flower', 'Easter Decoration'],  
'Dandelion': ['Dandelion Flower', 'Yellow Balloon'],  
'Dark Brown': ['Chocolate Cake', 'Dark Wood'],  
'Dark Candy Apple Red': ['Candy Apple', 'Red Car'],  
'Dark Cerulean': ['Deep Ocean', 'Navy Uniform'],  
'Dark Chestnut': ['Horse', 'Chestnut Tree'],  
'Dark Coral': ['Coral Reef', 'Tropical Fish'],  
'Dark Cyan': ['Teal Sofa', 'Aquarium'],  
'Dark Electric Blue': ['Electric Guitar', 'Neon Sign'],  
'Dark Imperial Blue': ['Royal Robe', 'Evening Sky'],  
'Dark Jungle Green': ['Forest', 'Jungle Camouflage'],  
'Dark Lava': ['Volcanic Rock', 'Obsidian'],  
'Dark Lavender': ['Lavender Sachet', 'Purple Blanket'],  
'Dark Liver': ['Liver', 'Dark Brown Sofa'],  
'Dark Magenta': ['Magenta Flower', 'Purple Scarf'],  
'Dark Medium Gray': ['Slate Roof', 'Stone Path'],  
'Dark Midnight Blue': ['Night Sky', 'Dark Denim'],  
'Dark Moss Green': ['Moss', 'Forest Canopy'],  
'Dark Orchid': ['Orchid Flower', 'Purple Velvet'],  
'Dark Pastel Green': ['Mint Leaf', 'Green Smoothie'],  
'Dark Red': ['Red Wine', 'Ruby'],  
'Dark Scarlet': ['Scarlet Robe', 'Red Curtain'],  
'Dark Sienna': ['Old Brick', 'Clay Pot'],  
'Dark Spring Green': ['Spring Grass', 'Leaf'],  
'Dark Tan': ['Leather Belt', 'Brown Jacket'],  
'Dark Tangerine': ['Tangerine', 'Orange Peel'],  
'Dark Terracotta': ['Terracotta Pot', 'Roof Tile'],  
'Dark Violet': ['Violet Flower', 'Purple Velvet'],  
'Jet Black': ['Night', 'Black hair'],  
'Ebony': ['Bones'],  
'Onyx': ['Gem'],  
'Obsidian': ['Gem'],  
'Dark Lava': ['Lava'],  
'Charcoal': ['Charcoal Briquette'],  
'Dark Gray': ['Gray Cloud'],  
'Light Gray': ['Gray Cloud'],  
'Outer Space': ['Outer Space'],  
'Old Silver': ['Coin'],  
'Khaki Drab': ['Dress Pants'],

```

'Desert Sand': ['Desert Dune'],
'Rosy Brown': ['Rosy Cheeks'],
'Shadow': ['Shadow'],
'Saddle Brown': ['Saddle Leather'],
'Chocolate': ['Chocolate Bar'],
'Peru': ['Rich Soil', 'Certain Types of Rock'],
'Apple Green': ['Apple', 'Green Bell Pepper'],
'Apricot': ['Apricot', 'Peach'],
'Eerie Black': ['Dark Forest'],
'Gun metal': ['Gun'],
'Davys Grey': ['Bass Fish'],
'Raisin Black': ['Raisin']
}

```

```
def get_color_makeup():
```

```

    return {
        'Red': ['Red-100%'],
        'Green': ['Green-100%'],
        'Blue': ['Blue-100%'],
        'Yellow': ['Yellow-100%'],
        'Cyan': ['Blue-50%', 'Green-50%'],
        'Magenta': ['Red-50%', 'Blue-50%'],
        'White': ['Red-33%', 'Green-33%', 'Blue-33%'],
        'Black': ['Black-100%'],
        'Gray': ['White-50%', 'Black-50%'],
        'Orange': ['Red-50%', 'Yellow-50%'],
        'Pink': ['Red-50%', 'White-50%'],
        'Purple': ['Red-50%', 'Blue-50%'],
        'Brown': ['Red-50%', 'Green-25%', 'Blue-25%'],
        'Maroon': ['Red-50%', 'Black-50%'],
        'Olive': ['Yellow-50%', 'Black-50%'],
        'Lime': ['Green-100%'],
        'Teal': ['Green-50%', 'Blue-50%'],
        'Navy': ['Blue-50%', 'Black-50%'],
        'Aquamarine': ['Green-50%', 'Blue-50%'],
        'Azure': ['Blue-50%', 'White-50%'],
        'Beige': ['Yellow-50%', 'White-50%'],
        'Bisque': ['Orange-50%', 'White-50%'],
        'Chocolate': ['Brown-100%'],
        'Coral': ['Red-50%', 'Orange-50%'],
        'Crimson': ['Red-60%', 'Blue-20%', 'Purple-20%'],
        'Fuchsia': ['Red-50%', 'Blue-50%'],
        'Gold': ['Yellow-70%', 'Orange-30%'],
        'Ivory': ['White-100%'],
        'Khaki': ['Yellow-70%', 'Green-30%'],
        'Lavender': ['Purple-70%', 'White-30%'],
        'Linen': ['White-80%', 'Yellow-20%'],
        'Moccasin': ['Yellow-50%', 'White-50%'],
        'Orchid': ['Purple-70%', 'Pink-30%'],
        'Peru': ['Orange-50%', 'Brown-30%', 'Red-20%'],
        'Plum': ['Purple-70%', 'Red-30%'],
        'Salmon': ['Red-60%', 'Orange-40%'],
        'Sienna': ['Brown-70%', 'Red-30%'],
        'Silver': ['Gray-100%'],
        'Tan': ['Brown-70%', 'Yellow-30%'],
        'Thistle': ['Purple-50%', 'White-50%'],
    }

```

'Tomato': ['Red-80%', 'Orange-20%'],  
'Turquoise': ['Blue-50%', 'Green-50%'],  
'Violet': ['Blue-50%', 'Red-50%'],  
'Wheat': ['Yellow-70%', 'White-30%'],  
'Sky Blue': ['Blue-70%', 'White-30%'],  
'Forest Green': ['Green-100%'],  
'Indian Red': ['Red-60%', 'Brown-40%'],  
'Steel Blue': ['Blue-70%', 'Gray-30%'],  
'Dark Slate Blue': ['Blue-70%', 'Gray-30%'],  
'Medium Spring Green': ['Green-80%', 'Yellow-20%'],  
'Rosy Brown': ['Brown-70%', 'Red-30%'],  
'Dark Olive Green': ['Green-70%', 'Yellow-30%'],  
'Saddle Brown': ['Brown-100%'],  
'Medium Purple': ['Purple-70%', 'Blue-30%'],  
'Medium Sea Green': ['Green-70%', 'Blue-30%'],  
'Fire Brick': ['Red-70%', 'Brown-30%'],  
'Cadet Blue': ['Blue-50%', 'Gray-50%'],  
'Pale Violet Red': ['Red-60%', 'Purple-40%'],  
'Dark Khaki': ['Yellow-50%', 'Green-30%', 'Red-20%'],  
'Light Coral': ['Red-70%', 'Orange-30%'],  
'Dark Salmon': ['Red-60%', 'Orange-40%'],  
'Medium Aquamarine': ['Blue-40%', 'Green-40%', 'Yellow-20%'],  
'Dark Orchid': ['Purple-70%', 'Red-30%'],  
'Medium Orchid': ['Purple-60%', 'Red-40%'],  
'Dark Turquoise': ['Blue-50%', 'Green-40%', 'Yellow-10%'],  
'Medium Turquoise': ['Blue-50%', 'Green-40%', 'Yellow-10%'],  
'Light Salmon': ['Red-50%', 'Orange-30%', 'Yellow-20%'],  
'Dark Goldenrod': ['Yellow-60%', 'Orange-30%', 'Red-10%'],  
'Medium Violet Red': ['Purple-50%', 'Red-50%'],  
'Light Green': ['Green-50%', 'Yellow-30%', 'Blue-20%'],  
'Light Sky Blue': ['Blue-70%', 'Green-20%', 'Yellow-10%'],  
'Light Steel Blue': ['Blue-60%', 'Green-20%', 'Yellow-20%'],  
'Pale Goldenrod': ['Yellow-50%', 'Green-30%', 'Red-20%'],  
'Light Yellow': ['Yellow-70%', 'Green-20%', 'Red-10%'],  
'Mint Cream': ['Green-50%', 'Yellow-30%', 'Blue-20%'],  
'Honeydew': ['Green-50%', 'Yellow-30%', 'Blue-20%'],  
'Alice Blue': ['Blue-70%', 'Green-20%', 'Yellow-10%'],  
'Ghost White': ['Blue-60%', 'Green-20%', 'Yellow-20%'],  
'Snow': ['White-100%'],  
'Seashell': ['White-80%', 'Pink-20%'],  
'Floral White': ['White-100%'],  
'Old Lace': ['White-100%'],  
'Antique White': ['White-80%', 'Yellow-20%'],  
'Papaya Whip': ['White-60%', 'Yellow-40%'],  
'Blanched Almond': ['White-50%', 'Yellow-30%', 'Red-20%'],  
'Lavender Blush': ['Purple-70%', 'White-30%'],  
'Cornsilk': ['Yellow-70%', 'White-30%'],  
'Lemon Chiffon': ['Yellow-70%', 'White-30%'],  
'Light Goldenrod Yellow': ['Yellow-70%', 'White-30%'],  
'Light Cyan': ['Cyan-100%'],  
'Pale Green': ['Green-70%', 'Yellow-20%', 'Blue-10%'],  
'Spring Green': ['Green-80%', 'Yellow-20%'],  
'Sea Green': ['Green-70%', 'Blue-20%', 'Yellow-10%'],  
'Dark Green': ['Green-100%'],  
'Yellow Green': ['Green-70%', 'Yellow-30%'],  
'Olive Drab': ['Green-50%', 'Yellow-40%', 'Red-10%'],  
'Dark Sea Green': ['Green-60%', 'Yellow-30%', 'Blue-10%'],



'Light Sea Green': ['Green-50%', 'Blue-30%', 'Yellow-20%'],  
'Light Goldenrod': ['Yellow-60%', 'White-40%'],  
'Goldenrod': ['Yellow-60%', 'Orange-30%', 'Red-10%'],  
'Rosy Brown': ['Brown-60%', 'Red-30%', 'White-10%'],  
'Burlywood': ['Brown-60%', 'Yellow-30%', 'White-10%'],  
'Navajo White': ['White-70%', 'Yellow-20%', 'Red-10%'],  
'Peach Puff': ['White-50%', 'Yellow-30%', 'Red-20%'],  
'Light Pink': ['Pink-70%', 'White-30%'],  
'Deep Pink': ['Pink-80%', 'Red-20%'],  
'Hot Pink': ['Pink-80%', 'Red-20%'],  
'Violet Red': ['Red-60%', 'Purple-40%'],  
'Medium Slate Blue': ['Blue-60%', 'Purple-40%'],  
'Cornflower Blue': ['Blue-70%', 'Purple-30%'],  
'Royal Blue': ['Blue-80%', 'Purple-20%'],  
'Dodger Blue': ['Blue-80%', 'Green-20%'],  
'Deep Sky Blue': ['Blue-80%', 'Green-20%'],  
'Light Blue': ['Blue-70%', 'Green-30%'],  
'Powder Blue': ['Blue-70%', 'Green-30%'],  
'Pale Turquoise': ['Blue-70%', 'Green-30%'],  
'Sky Blue': ['Blue-70%', 'Green-30%'],  
'Pale Goldenrod': ['Yellow-70%', 'White-30%'],  
'Peach Puff': ['White-50%', 'Yellow-30%', 'Red-20%'],  
'Pale Green': ['Green-70%', 'Yellow-20%', 'Blue-10%'],  
'Pale Turquoise': ['Blue-70%', 'Green-30%'],  
'Charcoal': ['Black-100%'],  
'Dark Gray': ['Gray-100%'],  
'Light Gray': ['Gray-100%'],  
'Ash Gray': ['Gray-70%', 'White-30%'],  
'Jet': ['Black-100%'],  
'Smoky Black': ['Black-100%'],  
'Outer Space': ['Black-80%', 'White-20%'],  
'Gun Metal': ['Black-70%', 'White-30%'],  
'Timberwolf': ['Gray-70%', 'White-30%'],  
'Platinum': ['Gray-80%', 'White-20%'],  
'Old Silver': ['Gray-70%', 'White-30%'],  
'Khaki Drab': ['Green-50%', 'Yellow-30%', 'Red-20%'],  
'Desert Sand': ['Yellow-70%', 'Brown-30%'],  
'Pale Silver': ['Gray-70%', 'White-30%'],  
'Shadow': ['Gray-60%', 'Black-40%'],  
'Pewter Blue': ['Blue-50%', 'Gray-50%'],  
'Cool Black': ['Black-100%'],  
'Cerulean': ['Blue-70%', 'Green-30%'],  
'Ultramarine': ['Blue-80%', 'Purple-20%'],  
'Electric Blue': ['Blue-60%', 'Green-40%'],  
'Jade': ['Green-70%', 'Blue-30%'],  
'Persian Blue': ['Blue-70%', 'Purple-30%'],  
'Cerulean Blue': ['Blue-70%', 'Green-30%'],  
'Egyptian Blue': ['Blue-70%', 'Green-30%'],  
'Klein Blue': ['Blue-70%', 'Purple-30%'],  
'Space Cadet': ['Blue-60%', 'Purple-40%'],  
'Cinereous': ['Gray-70%', 'White-30%'],  
'Cinnabar': ['Red-70%', 'Orange-30%'],  
'Cinnamon': ['Brown-70%', 'Red-30%'],  
'Citrine': ['Yellow-80%', 'Green-20%'],  
'Classic Rose': ['Pink-70%', 'White-30%'],  
'Cobalt Blue': ['Blue-80%', 'Purple-20%'],  
'Columbia Blue': ['Blue-70%', 'Green-30%'],

'Congo Pink': ['Pink-70%', 'White-30%'],  
'Copper': ['Brown-70%', 'Red-30%'],  
'Copper Rose': ['Brown-70%', 'Red-30%'],  
'Coquelicot': ['Red-70%', 'Orange-30%'],  
'Coral Pink': ['Pink-70%', 'White-30%'],  
'Cordovan': ['Brown-70%', 'Red-30%'],  
'Cornflower': ['Blue-70%', 'Purple-30%'],  
'Cosmic Cobalt': ['Blue-70%', 'Purple-30%'],  
'Cosmic Latte': ['White-100%'],  
'Cotton Candy': ['Pink-70%', 'White-30%'],  
'Cream': ['White-80%', 'Yellow-20%'],  
'Crimson Red': ['Red-80%', 'Purple-20%'],  
'Cyan Blue': ['Blue-60%', 'Green-40%'],  
'Cyber Grape': ['Purple-70%', 'Blue-30%'],  
'Cyber Yellow': ['Yellow-80%', 'Green-20%'],  
'Daffodil': ['Yellow-80%', 'Green-20%'],  
'Dandelion': ['Yellow-80%', 'Green-20%'],  
'Dark Brown': ['Brown-100%'],  
'Dark Candy Apple Red': ['Red-70%', 'Purple-30%'],  
'Dark Cerulean': ['Blue-70%', 'Green-30%'],  
'Dark Chestnut': ['Brown-70%', 'Red-30%'],  
'Dark Coral': ['Red-70%', 'Orange-30%'],  
'Dark Cyan': ['Green-70%', 'Blue-30%'],  
'Dark Electric Blue': ['Blue-70%', 'Green-30%'],  
'Dark Imperial Blue': ['Blue-70%', 'Purple-30%'],  
'Dark Jungle Green': ['Green-100%'],  
'Dark Lava': ['Brown-70%', 'Black-30%'],  
'Dark Lavender': ['Purple-70%', 'Blue-30%'],  
'Dark Liver': ['Brown-70%', 'Gray-30%'],  
'Dark Magenta': ['Purple-80%', 'Red-20%'],  
'Dark Medium Gray': ['Gray-100%'],  
'Dark Midnight Blue': ['Blue-70%', 'Black-30%'],  
'Dark Moss Green': ['Green-100%'],  
'Dark Pastel Green': ['Green-80%', 'Yellow-20%'],  
'Dark Red': ['Red-100%'],  
'Dark Scarlet': ['Red-80%', 'Purple-20%'],  
'Dark Sienna': ['Brown-100%'],  
'Dark Spring Green': ['Green-100%'],  
'Dark Tan': ['Brown-80%', 'Yellow-20%'],  
'Dark Tangerine': ['Orange-80%', 'Yellow-20%'],  
'Dark Terracotta': ['Brown-70%', 'Orange-30%'],  
'Dark Violet': ['Purple-100%'],  
'Jet Black': ['Black-100%'],  
'Ebony': ['Black-100%'],  
'Onyx': ['Black-100%'],  
'Obsidian': ['Black-100%'],  
'Dark Lava': ['Brown-70%', 'Black-30%'],  
'Apple Green': ['Green-80%', 'Yellow-20%'],  
'Apricot': ['Orange-70%', 'Yellow-30%'],  
'Aqua': ['Cyan-100%'],  
'Army Green': ['Green-80%', 'Black-20%'],  
'Asparagus': ['Green-70%', 'Yellow-30%'],  
'Atomic Tangerine': ['Orange-70%', 'Red-30%'],  
'Baker-Miller Pink': ['Pink-70%', 'Red-30%'],  
'Banana Yellow': ['Yellow-80%', 'Green-20%'],  
'Barn Red': ['Red-80%', 'Brown-20%'],  
'Battleship Grey': ['Gray-100%'],

'Bazaar': ['Brown-70%', 'Red-30%'],  
'Blizzard Blue': ['Blue-70%', 'White-30%'],  
'Blue Bell': ['Blue-70%', 'Purple-30%'],  
'Blue Green': ['Blue-70%', 'Green-30%'],  
'Blue Violet': ['Blue-70%', 'Purple-30%'],  
'Blush': ['Pink-70%', 'Red-30%'],  
'Bole': ['Brown-70%', 'Red-30%'],  
'Brick Red': ['Red-70%', 'Brown-30%'],  
'Bright Green': ['Green-100%'],  
'Bright Lilac': ['Purple-70%', 'Pink-30%'],  
'Bright Maroon': ['Red-70%', 'Purple-30%'],  
'Bright Navy Blue': ['Blue-70%', 'Black-30%'],  
'Bright Pink': ['Pink-100%'],  
'Bright Turquoise': ['Blue-70%', 'Green-30%'],  
'Bright Ube': ['Purple-70%', 'Pink-30%'],  
'Bright Yellow': ['Yellow-100%'],  
'Brilliant Rose': ['Pink-70%', 'Purple-30%'],  
'Brink Pink': ['Pink-70%', 'Red-30%'],  
'British Racing Green': ['Green-80%', 'Black-20%'],  
'Bronze': ['Brown-70%', 'Yellow-30%'],  
'Brown Sugar': ['Brown-70%', 'Yellow-30%'],  
'Bud Green': ['Green-80%', 'Yellow-20%'],  
'Buff': ['Yellow-70%', 'White-30%'],  
'Burgundy': ['Red-70%', 'Purple-30%'],  
'Byzantine': ['Purple-70%', 'Pink-30%'],  
'Byzantium': ['Purple-100%'],  
'Cambridge Blue': ['Blue-70%', 'Green-30%'],  
'Camel': ['Brown-70%', 'Yellow-30%'],  
'Capri': ['Blue-70%', 'Green-30%'],  
'Cardinal': ['Red-80%', 'Brown-20%'],  
'Caribbean Green': ['Green-80%', 'Blue-20%'],  
'Carmine': ['Red-80%', 'Purple-20%'],  
'Carnation Pink': ['Pink-70%', 'White-30%'],  
'Carnelian': ['Red-80%', 'Brown-20%'],  
'Carrot Orange': ['Orange-70%', 'Yellow-30%'],  
'Ceil': ['Blue-70%', 'White-30%'],  
'Celadon': ['Green-70%', 'White-30%'],  
'Celeste': ['Blue-70%', 'White-30%'],  
'Cerise': ['Pink-70%', 'Red-30%'],  
'Cerulean Frost': ['Blue-70%', 'Gray-30%'],  
'Champagne': ['Yellow-70%', 'White-30%'],  
'Cherry': ['Red-80%', 'Pink-20%'],  
'Chestnut': ['Brown-80%', 'Red-20%'],  
'China Pink': ['Pink-70%', 'Red-30%'],  
'Chinese Red': ['Red-80%', 'Brown-20%'],  
'Chocolate (Traditional)': ['Brown-100%'],  
'Cinereous': ['Gray-70%', 'White-30%'],  
'Cinnabar': ['Red-70%', 'Orange-30%'],  
'Cinnamon': ['Brown-70%', 'Red-30%'],  
'Citrine': ['Yellow-80%', 'Green-20%'],  
'Classic Rose': ['Pink-70%', 'White-30%'],  
'Cobalt Blue': ['Blue-80%', 'Purple-20%'],  
'Columbia Blue': ['Blue-70%', 'Green-30%'],  
'Congo Pink': ['Pink-70%', 'White-30%'],  
'Copper': ['Brown-70%', 'Red-30%'],  
'Copper Rose': ['Brown-70%', 'Red-30%'],  
'Coquelicot': ['Red-70%', 'Orange-30%'],

```

'Coral Pink': ['Pink-70%', 'White-30%'],
'Cordovan': ['Brown-70%', 'Red-30%'],
'Cornflower': ['Blue-70%', 'Purple-30%'],
'Cosmic Cobalt': ['Blue-70%', 'Purple-30%'],
'Cosmic Latte': ['White-100%'],
'Cotton Candy': ['Pink-70%', 'White-30%'],
'Cream': ['White-80%', 'Yellow-20%'],
'Crimson Red': ['Red-80%', 'Purple-20%'],
'Cyan Blue': ['Blue-60%', 'Green-40%'],
'Cyber Grape': ['Purple-70%', 'Blue-30%'],
'Cyber Yellow': ['Yellow-80%', 'Green-20%'],
'Daffodil': ['Yellow-80%', 'Green-20%'],
'Dandelion': ['Yellow-80%', 'Green-20%'],
'Dark Brown': ['Brown-100%'],
'Dark Candy Apple Red': ['Red-70%', 'Purple-30%'],
'Dark Cerulean': ['Blue-70%', 'Green-30%'],
'Dark Chestnut': ['Brown-70%', 'Red-30%'],
'Dark Coral': ['Red-70%', 'Orange-30%'],
'Dark Cyan': ['Green-70%', 'Blue-30%'],
'Dark Electric Blue': ['Blue-70%', 'Green-30%'],
'Dark Imperial Blue': ['Blue-70%', 'Purple-30%'],
'Dark Jungle Green': ['Green-100%'],
'Dark Lava': ['Brown-70%', 'Black-30%'],
'Dark Lavender': ['Purple-70%', 'Blue-30%'],
'Dark Liver': ['Brown-70%', 'Gray-30%'],
'Dark Magenta': ['Purple-80%', 'Red-20%'],
'Dark Medium Gray': ['Gray-100%'],
'Dark Midnight Blue': ['Blue-70%', 'Black-30%'],
'Dark Moss Green': ['Green-100%'],
'Dark Pastel Green': ['Green-80%', 'Yellow-20%'],
'Dark Red': ['Red-100%'],
'Dark Scarlet': ['Red-80%', 'Purple-20%'],
'Dark Sienna': ['Brown-100%'],
'Dark Spring Green': ['Green-100%'],
'Dark Tan': ['Brown-80%', 'Yellow-20%'],
'Dark Tangerine': ['Orange-80%', 'Yellow-20%'],
'Dark Terracotta': ['Brown-70%', 'Orange-30%'],
'Dark Violet': ['Purple-100%'],
'Gunmetal': ['Green-70%', 'Black-30%'],
'Eerie Black': ['Black-100%'],
'Davys Grey': ['Green-30%', 'Grey-70%'],
'Raisin Black': ['Black-100%']
}

```

## “Main.py”

```

import cv2
import numpy as np
from scipy.spatial import KDTree
import pyttsx3
from Utilpt2 import get_limits, get_color_bgr_values, get_color_objects, get_color_makeup

# Global variables
clicked_color = None

```

```

clicked_position = (-1, -1)
color_box_size = 150
show_objects = False
show_speaker = False
average_area_size = 15

# Extended list of colors and names
colors = get_color_bgr_values()

# Convert predefined colors to Lab space
lab_colors = {name: cv2.cvtColor(np.uint8([[value]]), cv2.COLOR_BGR2Lab)[0][0] for name, value in colors.items()}

# Create KDTree for fast nearest-neighbor lookup
lab_values = np.array(list(lab_colors.values()))
kdtree = KDTree(lab_values)

# Example objects associated with colors
objects = get_color_objects()

# Initialize text-to-speech engine
engine = pyttsx3.init()

def speak(text):
    engine.say(text)
    engine.runAndWait()

def get_color_name(bgr_color):
    clicked_lab = cv2.cvtColor(np.uint8([[bgr_color]]), cv2.COLOR_BGR2Lab)[0][0]
    _, index = kdtree.query(clicked_lab)
    color_name = list(lab_colors.keys())[index]
    return color_name

def get_color(event, _x, _y, flags, param):
    global clicked_color, clicked_position, show_objects, show_speaker
    if event == cv2.EVENT_LBUTTONDOWN:
        if 10 <= _x <= 80 and 90 <= _y <= 110:
            show_objects = not show_objects
            print("Objects text box clicked. Toggling show_objects to", show_objects)
        elif frame.shape[1] - 100 <= _x <= frame.shape[1] - 20 and frame.shape[0] - 50 <= _y <= frame.shape[0] - 30:
            show_speaker = not show_speaker
            print("Speaker text box clicked. Toggling show_speaker to", show_speaker)
            if show_speaker and clicked_color is not None:
                speak_color_info()
        else:
            clicked_position = (_x, _y)
            clicked_color = average_color(frame, clicked_position, average_area_size)
            show_objects = False
            show_speaker = False

```

```
print("Color selected at position:", clicked_position, "Average color:", clicked_color)
```

```
def speak_color_info():
    if clicked_color is not None:
        color_name = get_color_name(clicked_color)
        color_makeup = get_color_makeup().get(color_name, [])
        darkness_level = calculate_darkness(clicked_color)

        text = f"The selected color is {color_name}. "
        if color_makeup:
            percentages = ', '.join(f'{makeup}' for makeup in color_makeup)
            text += f"It consists of {percentages}. "
            text += f"The darkness level is {darkness_level:.1f} out of 10."

        if color_name in objects:
            object_names = ', '.join(objects[color_name])
            text += f" It is commonly associated with {object_names}."

    speak(text)
```

```
def average_color(image, position, size):
    x, y = position
    height, width, _ = image.shape
    x_start = max(x - size, 0)
    x_end = min(x + size, width)
    y_start = max(y - size, 0)
    y_end = min(y + size, height)
    area = image[y_start:y_end, x_start:x_end]

    # Apply Gaussian blur to the selected area
    blurred_area = cv2.GaussianBlur(area, (5, 5), 0)

    average_color = cv2.mean(blurred_area)[:3]
    return np.array(average_color, dtype=np.uint8)
```

```
def increase_saturation(frame, saturation_scale=1.2):
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    hsv = np.array(hsv, dtype=np.float64)
    hsv[:, :, 1] = hsv[:, :, 1] * saturation_scale
    hsv[:, :, 1][hsv[:, :, 1] > 255] = 255
    hsv = np.array(hsv, dtype=np.uint8)
    return cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
```

```
def draw_text_with_outline(image, text, position, font, scale, color, thickness, outline_color=(0, 0, 0),
                           outline_thickness=2):
    # Draw the outline
```

```

cv2.putText(image, text, position, font, scale, outline_color, outline_thickness, lineType=cv2.LINE_AA)
# Draw the text
cv2.putText(image, text, position, font, scale, color, thickness, lineType=cv2.LINE_AA)

```

```

def draw_centered_text(image, text, font, scale, color, thickness, outline_color=(0, 0, 0), outline_thickness=2,
                        max_width=None):
    if max_width is not None:
        text_size = cv2.getTextSize(text, font, scale, thickness)[0]
        while text_size[0] > max_width and scale > 0.1:
            scale -= 0.1
            text_size = cv2.getTextSize(text, font, scale, thickness)[0]

    text_size = cv2.getTextSize(text, font, scale, thickness)[0]
    text_x = (image.shape[1] - text_size[0]) // 2
    text_y = text_size[1] + 10
    draw_text_with_outline(image, text, (text_x, text_y), font, scale, color, thickness, outline_color,
                           outline_thickness)

```

```

def calculate_darkness(bgr_color):
    gray_color = cv2.cvtColor(np.uint8([bgr_color]), cv2.COLOR_BGR2GRAY)[0][0]
    return (255 - gray_color) / 255 * 10

```

```

cap = cv2.VideoCapture(0)
cv2.namedWindow('frame')
cv2.setMouseCallback('frame', get_color)

```

```

try:
    while True:
        ret, frame = cap.read()
        if not ret:
            print("Failed to capture image")
            break

        # Increase the saturation of the frame
        frame = increase_saturation(frame, saturation_scale=1.2)

```

```

    if clicked_color is not None:
        # Perform operations based on clicked color
        box_x = max(clicked_position[0] - 5, 0)
        box_y = max(clicked_position[1] - 5, 0)
        box_width = min(10, frame.shape[1] - box_x)
        box_height = min(10, frame.shape[0] - box_y)

        cv2.rectangle(frame, (box_x, box_y), (box_x + box_width, box_y + box_height), (0, 255, 0), 2)

        bgr_box = np.full((80, color_box_size, 3), clicked_color, dtype=np.uint8)
        color_name = get_color_name(clicked_color)
        draw_centered_text(bgr_box, color_name, cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 255, 255), 2,

```

```

        max_width=color_box_size)

color_makeup = get_color_makeup().get(color_name, [])
y_offset = 60
for makeup in color_makeup:
    text = f"{makeup}"
    draw_text_with_outline(bgr_box, text, (5, y_offset), cv2.FONT_HERSHEY_SIMPLEX, 0.4, (255, 255, 255), 1)
    y_offset += 15

frame[10:90, 10:10 + color_box_size] = bgr_box

# Draw the "Objects" text box
cv2.rectangle(frame, (10, 90), (80, 110), (255, 0, 0), -1)
draw_text_with_outline(frame, "Objects", (15, 105), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 1)

# Calculate and display darkness level
darkness_level = calculate_darkness(clicked_color)
darkness_text = "1 very light    10 very dark"
line_start = (frame.shape[1] - 290, 20)
line_end = (frame.shape[1] - 80, 20)
cv2.line(frame, line_start, line_end, (255, 255, 255), 1)
draw_text_with_outline(frame, darkness_text, (frame.shape[1] - 290, 10), cv2.FONT_HERSHEY_SIMPLEX, 0.4,
                       (255, 255, 255), 1)
cv2.circle(frame, (frame.shape[1] - 185, 50), 15, clicked_color.tolist(), -1)
draw_text_with_outline(frame, f"{darkness_level:.1f}", (frame.shape[1] - 195, 55), cv2.FONT_HERSHEY_SIMPLEX,
                       0.4, (255, 255, 255), 1)

# Draw the "Speaker" text box
cv2.rectangle(frame, (frame.shape[1] - 100, frame.shape[0] - 50),
              (frame.shape[1] - 20, frame.shape[0] - 30), (0, 0, 255), -1)
draw_text_with_outline(frame, "Speaker", (frame.shape[1] - 95, frame.shape[0] - 35),
                       cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 1)

# Draw the objects box if show_objects is True
if show_objects and clicked_color is not None:
    objects_box = np.zeros((80, color_box_size, 3), dtype=np.uint8)
    color_name = get_color_name(clicked_color)
    if color_name in objects:
        object_names = objects[color_name]
        y_offset = 20
        for obj_name in object_names:
            draw_text_with_outline(objects_box, obj_name, (5, y_offset), cv2.FONT_HERSHEY_SIMPLEX, 0.5,
                                  (255, 255, 255), 1)
            y_offset += 20
        frame[10:90, 200:200 + color_box_size] = objects_box

cv2.imshow('frame', frame)

if cv2.waitKey(1) & 0xFF == ord('q'):
    break

finally:

```



```
cap.release()  
cv2.destroyAllWindows()
```