```
import cv2
import numpy as np

def detect_color(image_path):
    # Read the image
    image = cv2.imread(image_path)

    # Convert image from BGR to RGB (OpenCV reads images in BGR format by default)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    # Define color boundaries in RGB format
    boundaries = [
        ([17, 15, 100], [50, 56, 200]),   # Red
        ([86, 31, 4], [220, 88, 50]),     # Blue
        ([25, 146, 190], [62, 174, 250]), # Yellow
    ]

    # Loop over the boundaries
    for (lower, upper) in boundaries:
        # Convert the boundaries to NumPy arrays
        lower = np.array(lower, dtype="uint8")
        upper = np.array(upper, dtype="uint8")

        # Find the colors within the specified boundaries and apply the mask
        mask = cv2.inRange(image, lower, upper)
        output = cv2.bitwise_and(image, image, mask=mask)

        # Show the images
        cv2.imshow("images", np.hstack([image, output]))
        cv2.waitKey(0)
```

```
# Path to the image
image_path = "example_image.jpg"

# Call the color detection function
detect_color(image_path)
```