```
# taking a picture to test the camera

import cv2
import numpy as np
import RPi.GPIO as GPIO
import time
import os

# initalize variables
code_running=True

# quit button
GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.IN, pull_up_down=GPIO.PUD_UP)

def GPIO17_call_back(channel):
    global code_running
    code_running=False

# led set up (second motor controller)
GPIO.setup(25,GPIO.OUT)
GPIO.setup(23,GPIO.OUT)
GPIO.setup(24,GPIO.OUT) # PWM A
p_led=GPIO.PWM(24,50)
p_led.start(0)

# red:
light_red1=np.array([0,51,20])
dark_red1=np.array([15,255,255])
light_red2=np.array([160,51,20])
dark_red2=np.array([180,255,255])

# white:
light_white=np.array([0,0,200])
dark_white=np.array([180,30,255])

# darks:
light_dark1=np.array([0,100,0])
dark_dark1=np.array([180,255,100])
light_dark2=np.array([110,150,0])
dark_dark2=np.array([160,255,100])

# lights:
light_light=np.array([0,30,175])
dark_light=np.array([180,80,200])
```

```
46
47  # colors:
48  light_color=np.array([0,100,100])
49  dark_color=np.array([180,200,200])
50
51  # led direction
52  GPIO.output(23,GPIO.LOW)
53  GPIO.output(25,GPIO.HIGH)
54
55  while(code_running):
56
57      cap = cv2.VideoCapture(0) #video capture source camera
58
59      CLOTHING_RED=False
60      CLOTHING_COLOR=False
61      CLOTHING_WHITE=False
62      CLOTHING_DARK=False
63      CLOTHING_LIGHT=False
64
65      # turn light on
66      p_led.start(100)
67      time.sleep(0.5)
68      print("light on")
69
70      ret,frame = cap.read()
71      print("picture taken")
72
73      # turn off light
74      time.sleep(0.5)
75      p_led.ChangeDutyCycle(0)
76
77      # convert RBG to HSV
78      hsv=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
79
80      # find colors
81      mask_red1=cv2.inRange(hsv,light_red1,dark_red1)
82      mask_red2=cv2.inRange(hsv,light_red2,dark_red2)
83      mask_white=cv2.inRange(hsv,light_white,dark_white)
84      mask_dark1=cv2.inRange(hsv,light_dark1,dark_dark1)
85      mask_dark2=cv2.inRange(hsv,light_dark2,dark_dark2)
86      mask_light=cv2.inRange(hsv,light_light,dark_light)
87      mask_color=cv2.inRange(hsv,light_color,dark_color)
88
89      # output
90      red1=cv2.bitwise_and(frame,frame,mask=mask_red1)
```

```
91      red2=cv2.bitwise_and(frame,frame,mask=mask_red2)
92      white=cv2.bitwise_and(frame,frame,mask=mask_white)
93      dark1=cv2.bitwise_and(frame,frame,mask=mask_dark1)
94      dark2=cv2.bitwise_and(frame,frame,mask=mask_dark2)
95      light=cv2.bitwise_and(frame,frame,mask=mask_light)
96      color=cv2.bitwise_and(frame,frame,mask=mask_color)
97
98      # Find area
99      area_red1 = 0
100     area_red2 = 0
101     area_white = 0
102     area_dark1 = 0
103     area_dark2 = 0
104     area_light = 0
105     area_color = 0
106     area_red = 0
107     area_dark = 0
108
109     # Remove noise
110     kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (3,3))
111     opening_red1 = cv2.morphologyEx(mask_red1, cv2.MORPH_OPEN, kernel, iterations=1)
112     opening_red2 = cv2.morphologyEx(mask_red2, cv2.MORPH_OPEN, kernel, iterations=1)
113     opening_white = cv2.morphologyEx(mask_white, cv2.MORPH_OPEN, kernel, iterations=1)
114     opening_dark1 = cv2.morphologyEx(mask_dark1, cv2.MORPH_OPEN, kernel, iterations=1)
115     opening_dark2 = cv2.morphologyEx(mask_dark2, cv2.MORPH_OPEN, kernel,
        iterations=1)
116     opening_light = cv2.morphologyEx(mask_light, cv2.MORPH_OPEN, kernel,
        iterations=1)
117     opening_color = cv2.morphologyEx(mask_color, cv2.MORPH_OPEN, kernel, iterations=1)
118
119
120     # Find contours
121     original = frame.copy()
122     cnts_red1 = cv2.findContours(opening_red1, cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)
123     cnts_red1 = cnts_red1[0] if len(cnts_red1) == 2 else cnts_red1[1]
124     cnts_red2 = cv2.findContours(opening_red2, cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)
125     cnts_red2 = cnts_red2[0] if len(cnts_red2) == 2 else cnts_red2[1]
126     cnts_white = cv2.findContours(opening_white, cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)
127     cnts_white = cnts_white[0] if len(cnts_white) == 2 else cnts_white[1]
128     cnts_dark1 = cv2.findContours(opening_dark1, cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)
129     cnts_dark1 = cnts_dark1[0] if len(cnts_dark1) == 2 else cnts_dark1[1]
```

```
130     cnts_dark2 = cv2.findContours(opening_dark2, cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)
131     cnts_dark2= cnts_dark2[0] if len(cnts_dark2) == 2 else cnts_dark2[1]
132     cnts_light = cv2.findContours(opening_light, cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)
133     cnts_light = cnts_light[0] if len(cnts_light) == 2 else cnts_light[1]
134     cnts_color = cv2.findContours(opening_color, cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)
135     cnts_color = cnts_color[0] if len(cnts_color) == 2 else cnts_color[1]
136
137
138     for c in cnts_red1:
139         area_red1 += cv2.contourArea(c)
140         cv2.drawContours(original,[c], 0, (0,0,0), 2)
141     for c in cnts_red2:
142         area_red2 += cv2.contourArea(c)
143         cv2.drawContours(original,[c], 0, (0,0,0), 2)
144     for c in cnts_white:
145         area_white += cv2.contourArea(c)
146         cv2.drawContours(original,[c], 0, (0,0,0), 2)
147     for c in cnts_dark1:
148         area_dark1 += cv2.contourArea(c)
149         cv2.drawContours(original,[c], 0, (0,0,0), 2)
150     for c in cnts_dark2:
151         area_dark2 += cv2.contourArea(c)
152         cv2.drawContours(original,[c], 0, (0,0,0), 2)
153     for c in cnts_light:
154         area_light += cv2.contourArea(c)
155         cv2.drawContours(original,[c], 0, (0,0,0), 2)
156     for c in cnts_color:
157         area_color += cv2.contourArea(c)
158         cv2.drawContours(original,[c], 0, (0,0,0), 2)
159
160
161     # combine red and darks
162     area_red=area_red1+area_red2
163     area_dark=area_dark1+area_dark2
164
165     # Determine dominant color
166     if area_red>area_white and area_red>area_dark and area_red>area_light and
        area_red>area_color:
167         print('ning smells like cherries')
168         CLOTHING_RED = True
169     elif area_white>area_red and area_white>area_dark and area_white>area_light and
        area_white>area_color:
```

```
170             print('ning smells like a cloud')
171             CLOTHING_WHITE = True
172         elif area_dark>area_red and area_dark>area_white and area_dark>area_light and
            area_dark>area_color:
173             print('ning smells like dark chocolate')
174             CLOTHING_DARK = True
175         elif area_light>area_dark and area_light>area_red and area_light>area_white and
            area_light>area_color:
176             print('ning smells like an LED')
177             CLOTHING_LIGHT = True
178         else:
179             print('ning smells like a lucky charms')
180             CLOTHING_COLOR = True
181
182
183         cv2.imshow('img1',frame)
184         cv2.imshow('white',white)
185         cv2.imshow('light',light)
186         cv2.imshow('color',color)
187
188         cv2.waitKey(0)
189         time.sleep(0.5)
190         cap.release()
191         cv2.destroyAllWindows()
192
193
194         if cv2.waitKey()==1:
195             break
196
197
198
199
```