Elevi: Diana Trif, Andreea Tomescu
IA3

Barcode Detection și Color Detection

Proiectul nostru este compus din 2 aplicații OpenCV în Python, prima detectând codurile de bare și codurile QR, iar a doua culoarea obiectelor și schimbarea filtrelor de culoare manual.

Aplicația 1 - Barcode Detection

Am folosit 3 librării: OpenCV, numpy și pyzbar(ne ajută să detectăm și să localizăm barcodes și QR codes, el ne va decoda informația despre ele).

Decode este funcția care ne ajută să detectăm codurile si ne va dispune și poziția lor. O să putem scana codurile cu ajutorul camerei web, iar pe ecran va apărea detecția acestora, cât și link-ul codului. Decodăm și mesajele pe care fiecare barcode și QR code le au în proveniența lor. Detecția se face cu ajutorul unui contur care accentuează codurile găsite și alături va apărea si link-ul acestora.

De asemenea, putem verifica și dacă codurile sunt autorizate sau nu iar pentru acest lucru avem un fișier text cu id-uri care sunt autorizate. O să citim din fișier id-urile și le vom compara cu codurile noastre. Dacă codul este autorizat va fi detectat cu verde iar în caz contrar el va avea culoarea roșie.

1. **Codul aplicației:**

   1.1 Codul pentru detectare:

   QrBarTest.py

```
import cv2
import numpy as np
from pyzbar.pyzbar import decode

#img = cv2.imread('1.png')
cap = cv2.VideoCapture(0)   // pentru a putea scana coduri direct prin camera web.
cap.set(3,640)
cap.set(4,480)
```

```
while True:

    success, img = cap.read()
    for barcode in decode(img):
            myData = barcode.data.decode('utf-8') // este o metoda generală care transforma în sprint
        print(myData)
          pts = np.array([barcode.polygon],np.int32)// pune valorile poligonului în vector pentru a defini clar conturul detectării și localizarea ei prin contur
        pts = pts.reshape((-1,1,2))
              cv2.polylines(img,[pts],True,(255,0,255),5) // culoarea și grosimea conturului la detectare
        pts2 = barcode.rect // ne oferă poziția si informații adiționale

cv2.putText(img,myData,(pts2[0],pts2[1]),cv2.FONT_HERSHEY_SIMPLEX,
                0.9,(255,0,255),2)

    cv2.imshow('Result',img)
    cv2.waitKey(1)
```

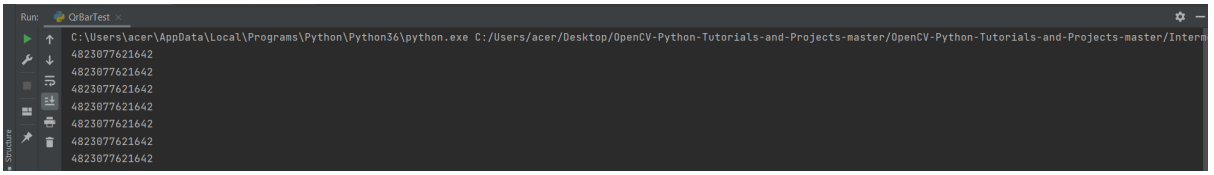
1.2. Codul pentru verificare autorizare:

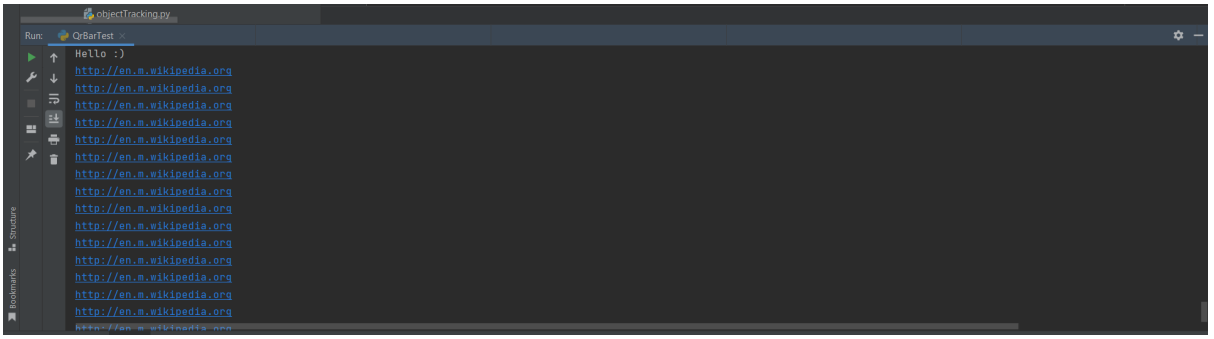
QrCodeProject.py

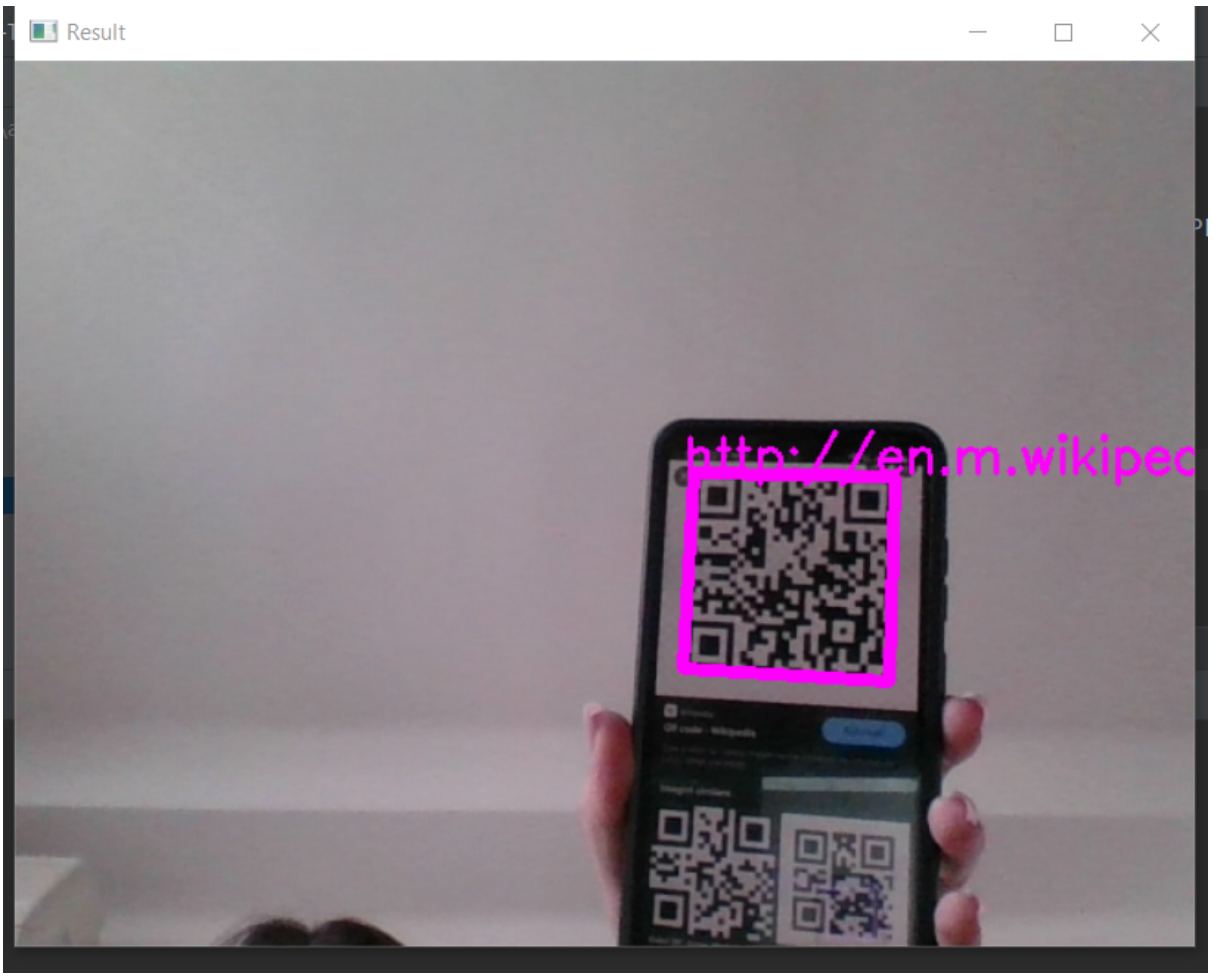
```
import cv2
import numpy as np
from pyzbar.pyzbar import decode

#img = cv2.imread('1.png')
cap = cv2.VideoCapture(0)// ne folosim de video
cap.set(3,640)
cap.set(4,480)

with open('myDataFile.text') as f:
   myDataList = f.read().splitlines()
// aici se citesc id-urile din fișierul txt

// aici are loc verificarea și detectarea personalizată în cazul celor 2 variante
while True:
```

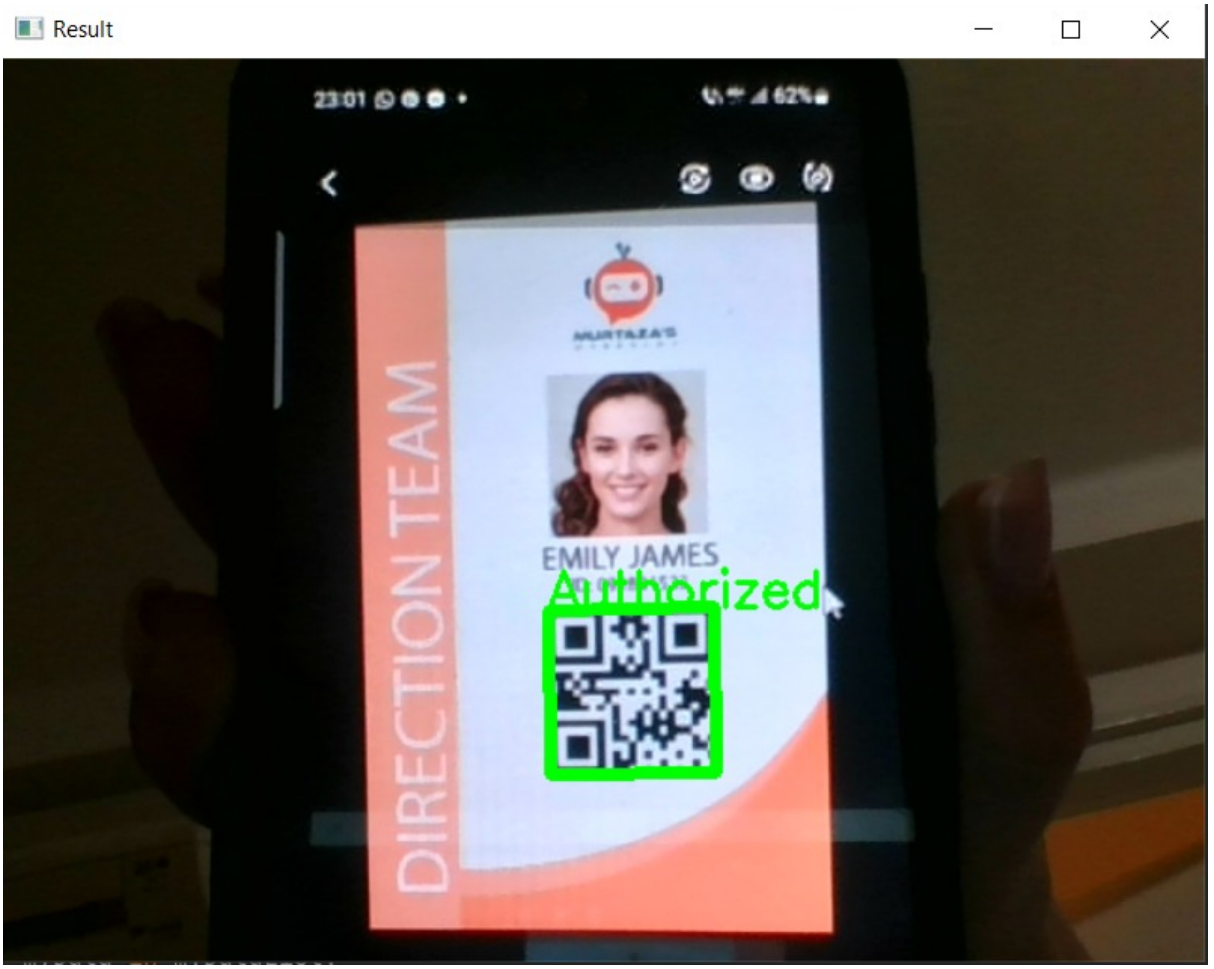
```
success, img = cap.read()
for barcode in decode(img):
    myData = barcode.data.decode('utf-8')
    print(myData)

    if myData in myDataList:
        myOutput = 'Authorized'
        myColor = (0,255,0)
    else:
        myOutput = 'Un-Authorized'
        myColor = (0, 0, 255)

    pts = np.array([barcode.polygon],np.int32)
    pts = pts.reshape((-1,1,2))
    cv2.polylines(img,[pts],True,myColor,5)
    pts2 = barcode.rect

cv2.putText(img,myOutput,(pts2[0],pts2[1]),cv2.FONT_HERSHEY_SIMPLEX,
                0.9,myColor,2)

cv2.imshow('Result',img)
cv2.waitKey(1)
```

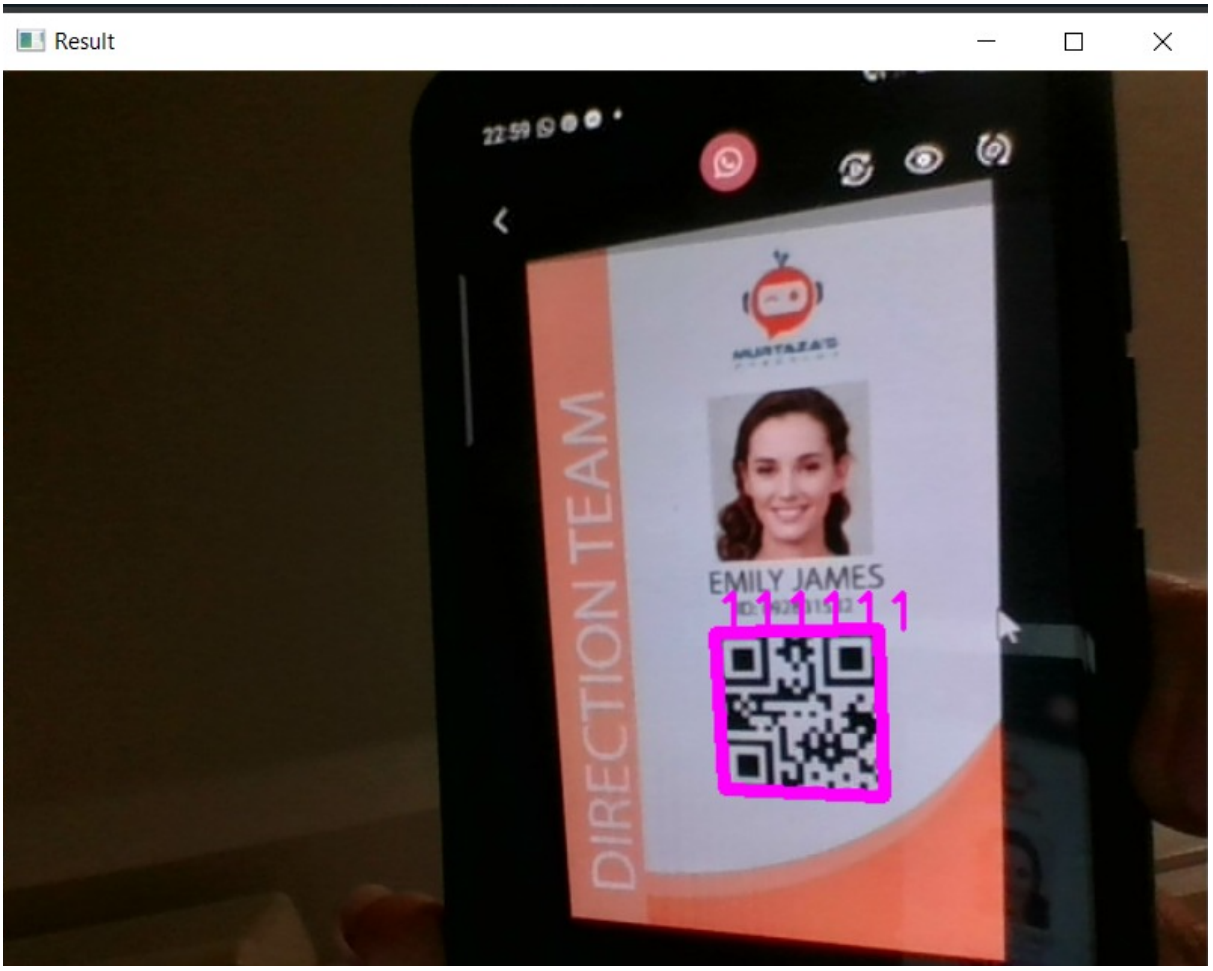
2. Date de ieșire






Result
http://en.m.wikipec


Result
23:01
62%
DIRECTION TEAM
EMILY JAMES
Authorized



Aplicația 2 - Color Detection

Aplicația arată cum putem detecta orice culoare dintr-o imagine folosind HSV cu ajutorul Opencv Track Bars. În exemplul nostru detectăm culoarea portocalie a mașinii în 3 pași: primul schimbăm culorile folosind HSV accentuând mașina, apoi culorile vor fi schimbate în alb negru(orice obiect care nu este folositor o sa fie negru iar mașina portocalie devine albă) ca mai apoi, în ultimul pas mașina să aibă culoarea originală pe fundal negru, astfel detectând culoarea mașinii.

1. **Codul aplicației:**

   detection.py

```
import cv2
import numpy as np

def empty(a):
```

```
  pass

def stackImages(scale,imgArray):
  rows = len(imgArray)
  cols = len(imgArray[0])
  rowsAvailable = isinstance(imgArray[0], list)
  width = imgArray[0][0].shape[1]
  height = imgArray[0][0].shape[0]
  if rowsAvailable:
    for x in range ( 0, rows):
      for y in range(0, cols):
        if imgArray[x][y].shape[:2] == imgArray[0][0].shape [:2]:
            imgArray[x][y] = cv2.resize(imgArray[x][y], (0, 0), None, scale,
scale)
        else:
                                imgArray[x][y] = cv2.resize(imgArray[x][y],
(imgArray[0][0].shape[1], imgArray[0][0].shape[0]), None, scale, scale)
          if len(imgArray[x][y].shape) == 2: imgArray[x][y]= cv2.cvtColor(
imgArray[x][y], cv2.COLOR_GRAY2BGR)
    imageBlank = np.zeros((height, width, 3), np.uint8)
    hor = [imageBlank]*rows
    hor_con = [imageBlank]*rows
    for x in range(0, rows):
      hor[x] = np.hstack(imgArray[x])
    ver = np.vstack(hor)
  else:
    for x in range(0, rows):
      if imgArray[x].shape[:2] == imgArray[0].shape[:2]:
        imgArray[x] = cv2.resize(imgArray[x], (0, 0), None, scale, scale)
      else:
            imgArray[x] = cv2.resize(imgArray[x], (imgArray[0].shape[1],
imgArray[0].shape[0]), None,scale, scale)
                        if len(imgArray[x].shape) == 2: imgArray[x] =
cv2.cvtColor(imgArray[x], cv2.COLOR_GRAY2BGR)
    hor= np.hstack(imgArray)
    ver = hor
  return ver

// aici se schimbă filtrele de culoare pentru detectarea culorii mașinii
```

```
path = 'Resources/lambo.png'
cv2.namedWindow("TrackBars")
cv2.resizeWindow("TrackBars",640,240)
cv2.createTrackbar("Hue Min","TrackBars",0,179,empty)
cv2.createTrackbar("Hue Max","TrackBars",19,179,empty)
cv2.createTrackbar("Sat Min","TrackBars",110,255,empty)
cv2.createTrackbar("Sat Max","TrackBars",240,255,empty)
cv2.createTrackbar("Val Min","TrackBars",153,255,empty)
cv2.createTrackbar("Val Max","TrackBars",255,255,empty)

while True:
  img = cv2.imread(path)
  imgHSV = cv2.cvtColor(img,cv2.COLOR_BGR2HSV)
  h_min = cv2.getTrackbarPos("Hue Min","TrackBars")
  h_max = cv2.getTrackbarPos("Hue Max", "TrackBars")
  s_min = cv2.getTrackbarPos("Sat Min", "TrackBars")
  s_max = cv2.getTrackbarPos("Sat Max", "TrackBars")
  v_min = cv2.getTrackbarPos("Val Min", "TrackBars")
  v_max = cv2.getTrackbarPos("Val Max", "TrackBars")
  print(h_min,h_max,s_min,s_max,v_min,v_max)
  lower = np.array([h_min,s_min,v_min])
  upper = np.array([h_max,s_max,v_max])
  mask = cv2.inRange(imgHSV,lower,upper)
  imgResult = cv2.bitwise_and(img,img,mask=mask)



  # cv2.imshow("Original",img)
  # cv2.imshow("HSV",imgHSV)
  # cv2.imshow("Mask", mask)
  # cv2.imshow("Result", imgResul

  imgStack = stackImages(0.6,([img,imgHSV],[mask,imgResult]))
  cv2.imshow("Stacked Images", imgStack)

  cv2.waitKey(1)
```

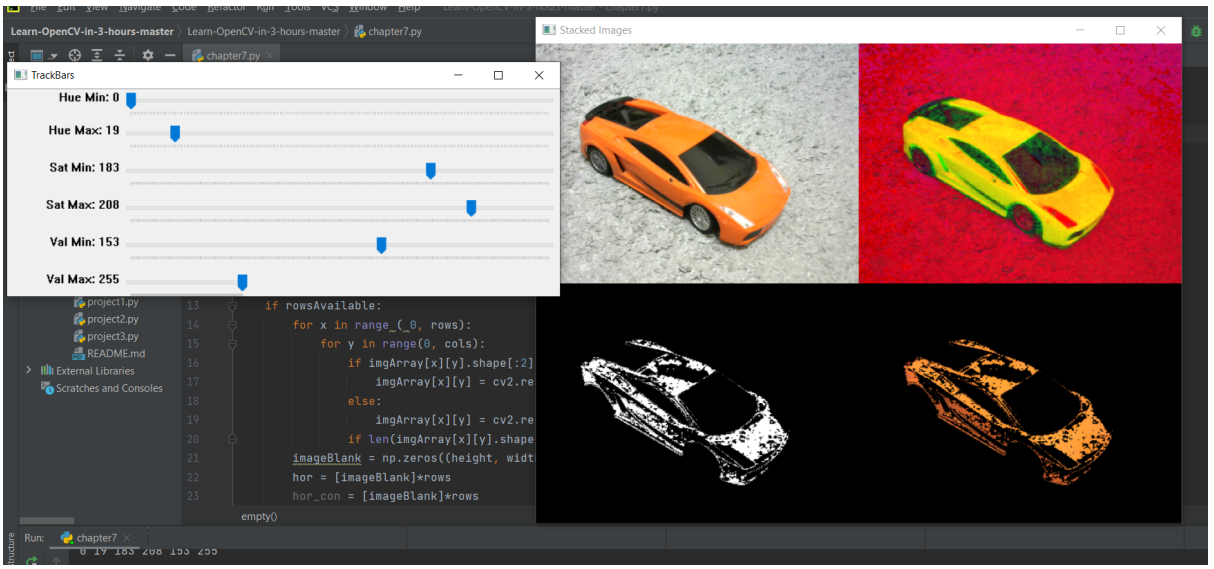
colorPicker.py

```
import cv2
import numpy as np

frameWidth = 640
frameHeight = 480
cap = cv2.VideoCapture(1)
cap.set(3, frameWidth)
cap.set(4, frameHeight)
cap.set(10,150)

def empty(a):
  pass

cv2.namedWindow("HSV")
cv2.resizeWindow("HSV",640,240)
cv2.createTrackbar("HUE Min","HSV",0,179,empty)
cv2.createTrackbar("SAT Min","HSV",0,255,empty)
cv2.createTrackbar("VALUE Min","HSV",0,255,empty)
cv2.createTrackbar("HUE Max","HSV",179,179,empty)
cv2.createTrackbar("SAT Max","HSV",255,255,empty)
cv2.createTrackbar("VALUE Max","HSV",255,255,empty)

while True:

  _, img = cap.read()
  imgHsv = cv2.cvtColor(img,cv2.COLOR_BGR2HSV)

  h_min = cv2.getTrackbarPos("HUE Min","HSV")
  h_max = cv2.getTrackbarPos("HUE Max", "HSV")
  s_min = cv2.getTrackbarPos("SAT Min", "HSV")
  s_max = cv2.getTrackbarPos("SAT Max", "HSV")
  v_min = cv2.getTrackbarPos("VALUE Min", "HSV")
  v_max = cv2.getTrackbarPos("VALUE Max", "HSV")
  print(h_min)

  lower = np.array([h_min,s_min,v_min])
  upper = np.array([h_max,s_max,v_max])
  mask = cv2.inRange(imgHsv,lower,upper)
  result = cv2.bitwise_and(img,img, mask = mask)
```

```
    mask = cv2.cvtColor(mask, cv2.COLOR_GRAY2BGR)
    hStack = np.hstack([img,mask,result])
    #cv2.imshow('Original', img)
    #cv2.imshow('HSV Color Space', imgHsv)
    #cv2.imshow('Mask', mask)
   #cv2.imshow('Result', result)
    cv2.imshow('Horizontal Stacking', hStack)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

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

2. Date de intrare

3. Date de ieșire



Link GitHub:https://github.com/DianaTrif00/Vedere-Artificiala

Bibliografie:

https://www.youtube.com/watch?v=SrZuwM705yE&list=PLMoSUbG1Q_r_sc0x7ndCsqdIkL7dwrmNF&index=17