**Virtual painting -VA**

Adam Ana, sg1, an 3, IA, FMI

Virtual painting este locul în care putem desena doar prin capturarea mișcareii mâinii cu o cameră. Noi suntem aici acum, folosind tehnici de viziune computerizată în open cv pentru a construi acest proiect. Limbajul necesar pentru acest proiect este python 3.6 si bibliotecile sale ușor de utilizat sintaxa, dar înțelegerea elementelor de bază. Urmărirea culorilor și procesele de detectare sunt utilizate pentru atingerea scopului acestui proiect.

Cuvinte cheie: open CV, Python, virtual painting.

Viziunea computerizată este o știință care învață computerele să “vadă”. Cu algoritmi de ultimă generație, această tehnologie se află în spatele multor aplicații, cum ar fi [mașinile cu conducere autonomă](https://analyticsindiamag.com/looking-beyond-self-driving-cars-how-ai-is-transforming-the-automotive-industry/) , [recunoașterea imaginilor,](https://analyticsindiamag.com/hands-on-guide-to-implement-deep-autoencoder-in-pytorch-for-image-reconstruction/) diagnosticarea medicală etc. Cea mai bună parte a vederii computerizate este că aceste tehnici sunt folosite pentru detectarea celulelor canceroase, ceea ce ajută la salvarea de vieți prin adăugarea de filtre. la fața ta pentru a te distra. În mod similar, are o mare varietate de aplicații interesante și utile.

În acest articol, vom realiza o implementare practică a modului de a crea o aplicație de pictură virtuală fără a atinge tastatura și doar desenând în aer care va fi afișat pe ecran.

Aceasta aplicatie este destinata tuturor persoanelor cu scopul de a se distra ,relaxa si a crea un mediu cat mai interesant

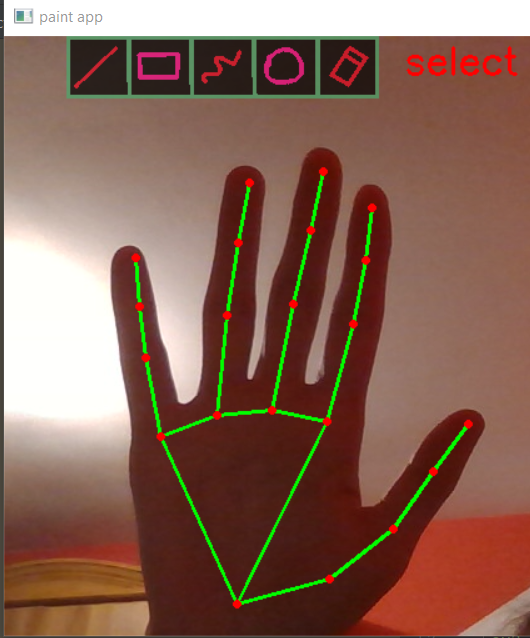
### ****Pași pentru această implementare:****

1. Import de biblioteci, alocarea valorilor BGR pentru culoare.
2. Crearea unei ferestre care oferă diverse opțiuni.
3. Capturarea unui videoclip folosind o cameră web și preluarea cadrelor.
4. Folosind diferite tehnici de desenare
5. Afișare output
6. Salvarea outputului

**1.Importarea bibliotecilor :**

import mediapipe as mp  
import cv2  
import numpy as np  
import time

* mediapipe-aceasta librarie ne vine in ajutor cu un algoritm,in spate, de detectie a mainilor,pentru a putea alege optiunile din meniu, cum ar fi: hand\_landmark-utilizat pentru a detecta linia mainii.



* Cv2: o bibliotecă de legături Python concepute pentru a rezolva problemele de vedere pe computer,cum ar fi importul / exportul unor imagini

tools = cv2.imread("tools.png")-citim imagine ape care vrem sa o utilizam

tools = tools.astype('uint8')

cv2.imwrite('C:/Users/alina/Desktop/img.jpg', frm)-salvam in PC-ul nostru, intr-o locatie dorita imaginea outputului

* numpy in python este imprtata pentru lurul cu matrici mult mai rapid decat o lista traditionala

tools = tools.astype('uint8')

* time- l am folosit ca sa determine durata unei epoci,cum ar fi inceputul si sfarsitul pentru a realiza desenul folosind un anumit tool
* if time\_init:  
   ctime = time.time()  
   time\_init = False

**2.Crearea unei ferestre care oferă diverse opțiuni.**

Pentru inceput am desenat in paint imaginea pentru instrumentele de desenat



Pe aceasta am adus-o in cod astfel:

tools = cv2.imread("tools.png")

Spatiul alocat fiecarei actiuni l-am alocat pe pixeli folosind functia “getTool” astfel:

if x < 50 + ml:  
 return "line"  
  
elif x < 100 + ml:  
 return "rectangle"  
  
elif x < 150 + ml:  
 return "draw"  
  
elif x < 200 + ml:  
 return "circle"  
  
else:  
 return "erase"

-pentru fiecare tool m-am gandit sa aloc 50 de pixeli

**3.Capturarea unui videoclip folosind o cameră web și preluarea cadrelor.**

Pentru acest pas m-am folosit de libraria cv2 pentru a porni camera ca sa putem lansa proiectul

cap = cv2.VideoCapture(0)

**4. Folosind diferite tehnici de desenare optiuni**

* Setare dimensiuni fereastra:

x, y = int(i.landmark[8].x \* 640), int(i.landmark[8].y \* 480)

* Optiunea “draw”,desenare libera-se recunoaste mana si apoi se aloca pixeli de le un punct la altul

cv2.line(mask, (prevx, prevy), (x, y), 0, thick)

* Optiunea “line”, linie-incepe de la 50 de pixeli

cv2.line(frm, (xii, yii), (x, y), (900, 152, 255), thick)

* Optiunea”rectangle”-desenare dreptunghi/ patrat
* Optiunea”circle”: am folosit formula matematica astfel:

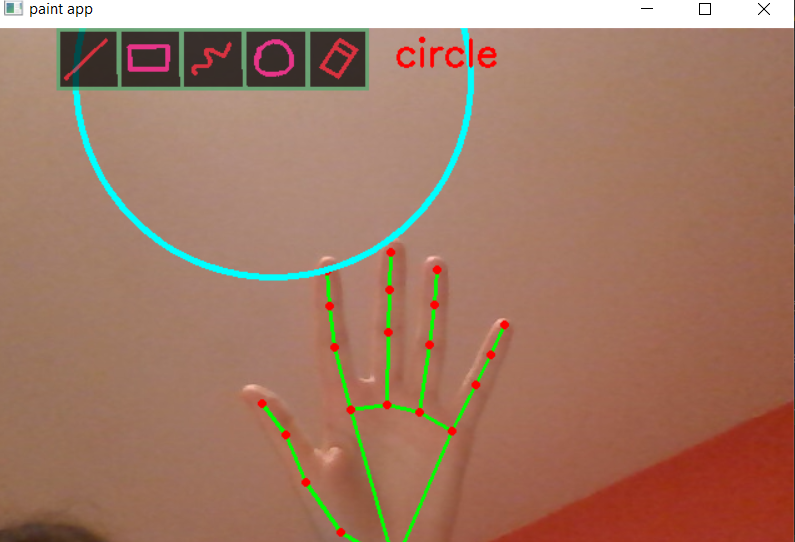
cv2.circle(mask, (xii, yii), int(((xii - x) \*\* 2 + (yii - y) \*\* 2) \*\* 0.5), (0, 255, 0), thick)

* Optiunea “erase”-stergere a desenului

Determinarea epocii de inceput si final: op = cv2.bitwise\_and(frm, frm, mask=mask)#determina epoca  
frm[:, :, 1] = op[:, :, 1]#incepe  
frm[:, :, 2] = op[:, :, 2]#se termina

**5.Afișare output**

Asa arata aplicatia odata porinta



**6.Salvarea outputului**

Pasul acesta se va salva la locatia dorita .

