-Hadirca Dionisie-

Proiect Computer Vision
Scrabble score calculator

Documentatie

1.Setup


Cod -> folderul care contine fisierul main.ipynb care ne rezolva problema

Images -> folderul care contine imaginile cu jocurile jucate (cerinta)

Imagini_auxiliare -> folderul care contine imaginile auxiliare date in cerinta
(avem nevoie de el pentru a extrage literele)

Predictions -> folderul care va contine fisierele ".txt" generate de program
"predictions/331_Hadirca_Dionisie/" (de la inceput e gol)

Letters -> un folder care va aparea pe parcursul rularii programului, vor fi literele decupate
din folderul cu imagini_auxiliare, utilizate la determinarea literii si calcularea scorului

2. Rulare si librarii folosite

```
In [6]: import cv2 as cv
        import numpy as np
        import string
        import os
        from copy import deepcopy

        print(np.version.version)
        print(cv.__version__)
```

```
1.23.3
4.6.0
```

Pentru a rula Programul trebuie sa rulati fiecare celula din fisierul main.ipynb consecutiv

3. Cum functioneaza?

Pasul 1: Extragem din folderul cu imagini pe rand fiecare imagine, si o trecem prin functia extract_grid(). Aceasta functie cuprinde mai multe modificari ale imaginii, precum filtrari, pentru a evidentia contururile. Dupa aplicam cateva functii implementate in libraria opencv:
a)threshold pentru a ramane cu imaginea si detaliile importante cum ar fi contururile boardului in format binar (alb/negru)
b)dilate pentru a evidentia contururile
c)Canny pentru a extrage muchiile
d)dilate pentru a evidentia contururile (iarasi)
e)functia findContours ne intoarce mai multe seturi de contururi care formeaza suprafete inchise, noi trebuie sa il gasim suprafata cu cel mai mare contur, ala va fi board-ul nostru

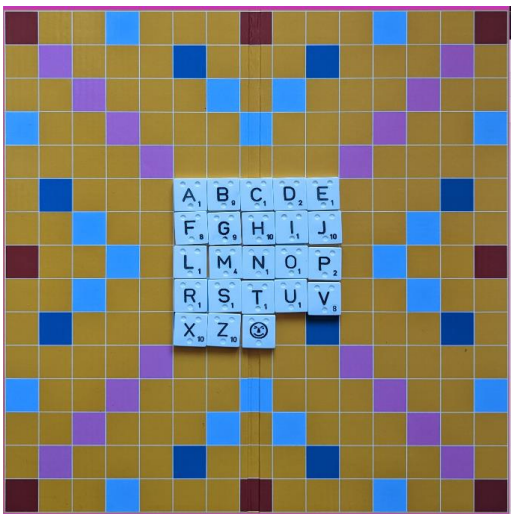
```
def extract_grid(image):
    work_image = cv.cvtColor(image,cv.COLOR_BGR2GRAY)
    image_m_blur = cv.medianBlur(work_image,3)
    image_g_blur = cv.GaussianBlur(image_m_blur,(0,0),5)
    image_sharpened = cv.addWeighted(image_m_blur,1.5,image_g_blur,-0.8,0)

    _,thresh = cv.threshold(image_sharpened,130,255,cv.THRESH_BINARY)

    kernel = np.ones((5, 5), np.uint8)
    thresh = cv.dilate(thresh, kernel)

    edges = cv.Canny(thresh,1,120)
    edges = cv.dilate(edges,kernel)

    contours, _ =cv.findContours(edges, cv.RETR_TREE, cv.CHAIN_APPROX_SIMPLE)

    image_copy = deepcopy(image)
    cv.drawContours(image_copy, contours, -1, (0,255,0), 3)
```

f) Am obtinut contururile, acum trebuie sa gasim punctele extreme care cuprind tabla de joc dupa care stocam imaginea obtinuta intr-o variabila cu informatii

```
image_copy = deepcopy(image)
cv.circle(image_copy,tuple(top_left),20,(0,0,255),-1)
cv.circle(image_copy,tuple(top_right),20,(0,0,255),-1)
cv.circle(image_copy,tuple(bottom_left),20,(0,0,255),-1)
cv.circle(image_copy,tuple(bottom_right),20,(0,0,255),-1)

width = NO_CELLS * CELL_WIDTH
height = NO_CELLS * CELL_WIDTH

puzzle = np.array([top_left,top_right,bottom_right,bottom_left], dtype = "float32")
destination_of_puzzle = np.array([[0,0],[width,0],[width,height],[0,height]], dtype = "float32")
M = cv.getPerspectiveTransform(puzzle,destination_of_puzzle)
result = cv.warpPerspective(image, M, (width, height))

return result
```

Prin Urmare:


devine =>



Pasul 2: Identificam pe ce pozitie a fost juacta o piesa. Pentru a realiza acest lucru am parcurs fiecare imagine obtinuta in urma pasului 1, si am verificat daca interiorul unei celule contine o singura culoare sau mai multe aplicand un threshold binar, apoi calculand media pe celula

Exemplu celula care contie piesa:

Exemplu celula care nu contine piesa:

Pasul 3: Identificam ce piesa a fost jucata. Pentru a realiza acest lucru am scris o functie care decupeaza literele din imagini auxiliare si le salveaza in folderul cu numele “letters”, dupa care am folosit o functie de template matching implementata in libraria openCV (matchTemplates()) care imi compara 2 imagini si intoarce similaritatea dintre ele.. Cu alte cuvinte simbolul jucat este simbolul care intoarce cel mai mare scor cu simbolul respectiv din folderul letters trecuta prin functia matchTemplates().

Exemplu litera obtinuta:

Exemplu litera de comparat

Putin work-around asupra sharpness-ului si niste filtrari, asupra ambelor litere imbunatatesc acuratetea

Pasul 4: Calcularea scorului: Pentru fiecare litera jucata trebuie sa determini ce cuvinte noi formeaza si sa calculezi scorul care il aduce acea litera, care mai apoi se aduna la scorul total pentru imaginea respectiva (bfs stanga-dreapta, sus-jos)

Pentru imaginile propuse in folderul de train programul meu a luat 100% din punctaj

```
Image:  5_19 Points position:  0.05 Points letters:  0.02 Points score:  0.02
Image:  5_20 Points position:  0.05 Points letters:  0.02 Points score:  0.02
8.99999999999995
```