

Documentatia pentru tema 1 – Calculator automat de scor pentru jocul Qwirkle

Fisierul solution.py implementeaza o solutie completa de procesare a imaginilor si de calculare a scorului pentru jocul Qwirkle Connect.

Pasii codului pentru fiecare imagine(runda) dintr-un joc:

1. Extragerea tablei din imagine
2. Detectarea pozitiilor cu puncte bonus (doar in prima imagine)
3. Detectarea pieselor de pe tabla
 - 3.1. Detectarea formelor pieselor
 - 3.2. Detectarea culorilor pieselor
4. Calcularea scorului
5. Scrierea rezultatului in fisier .txt

1. Extragerea tablei din imagine

Inainte de a afla informatii despre piesele de pe tabla si de a le prelucra, este nevoie de o vizualizare clara a tablei, iar acest lucru este posibil apeland functia **extract_square**.

Aceasta functie redimensioneaza imaginea initiala, aplica peste ea o masca HSV, cu valorile lower (28, 21, 0) si upper (151, 132, 255), ce evidentaiza conturul tablei si elimina fundalul, ca in figura 1. Apoi, functia aplica operatii morfologice pe imaginea rezultata pentru a defini mai bine contururile si pentru a elimina zgometul ramas. La final, se determina colturile tablei si se “extrag” tabla din imaginea initiala.

Astfel, imaginea rezultata este de 1280x1280 pixeli si va fi impartita ulterior in 32 de linii si 32 de coloane, fiecare “patratel” avand 80x80 pixeli.

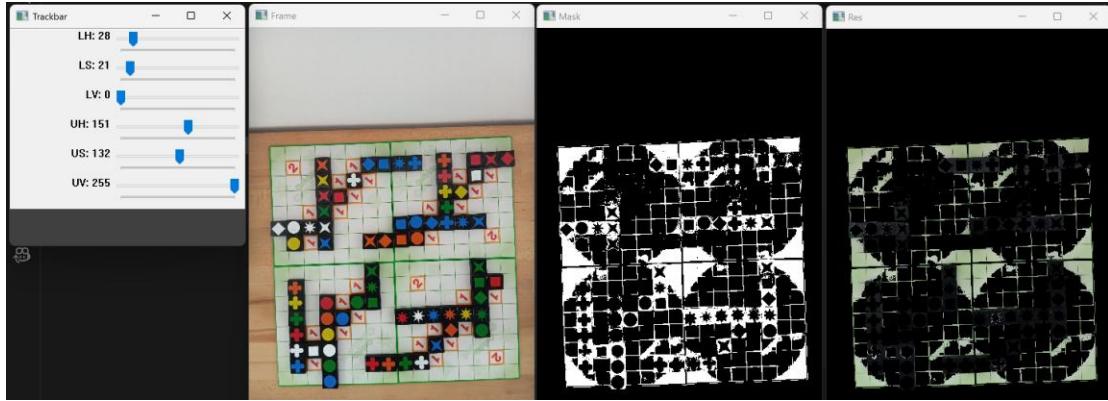


Figura 1

2. Detectarea pozitiilor cu puncte bonus

Acum ca imaginea citita a fost taiata pentru a avea doar tabla in ea, se poate face parcurgerea patratel cu patratel.

Punctele bonus vor fi cautate doar in imaginile ”_00.jpg”, caci punctele au aceeasi pozitie in fiecare imagine din cadrul unui joc si, astfel, nu este nevoie sa fie cautate la fiecare pas. Pentru a verifica daca un patratel este zona cu punct bonus, se aplica peste el o masca HSV cu valorile lower (0, 147, 160) si upper (16, 255, 255), ce detecteaza zonele cu puncte bonus, ca in figura 2, si, cu masca aplicata, apelam functia **crop_to_shape** pentru a mai taia din patratel pana cand avem continutul posibilei cifre pe intreaga suprafata. La final, se apeleaza functia **find_digit** pentru a identifica daca exista cifra in patratel si daca este 1 sau 2.

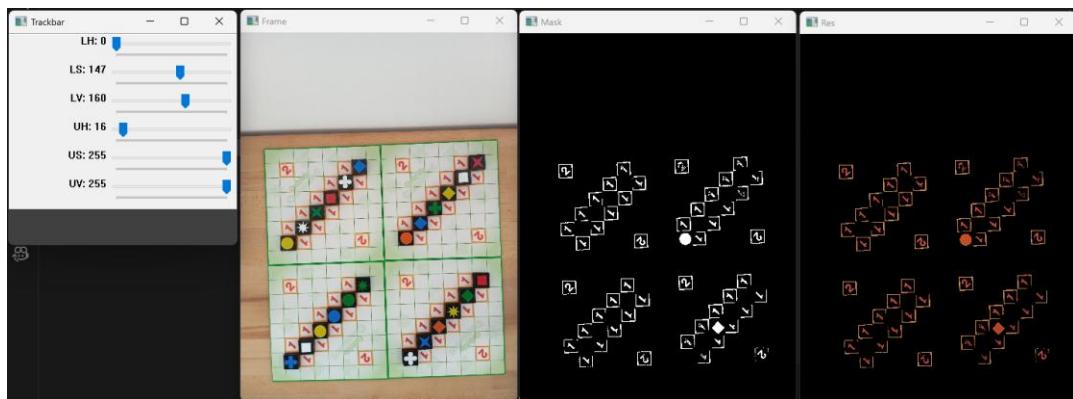


Figura 2

Functia **find_digit** foloseste template matching cu imagini din directorul templates, specific pentru cifrele 1 si 2. Se ia fiecare template, se pune peste patratel si se roteste la 90 de grade pana cand se gaseste potrivirea cea mai buna. Daca functia **find_digit** nu gaseste nicio potrivire, aceasta returneaza 0 si, in acest caz, verificam daca patratelul contine vreo piesa.

3. Detectarea pieselor de pe tabla

Pentru inceput, se aplica peste patratel o masca HSV cu valorile lower (0, 0, 66) si upper (184, 255, 255), ce identifica doar piesele de pe tabla, ca in figura 3. Daca se gaseste o piesa in patratel, se salveaza in lista shapes coordonatele patratelului doar in cazul in care piesa este gasita pentru prima data. Folosind patratelul cu masca aplicata, se apeleaza functia **crop_to_shape** pentru a taia fundalul din patratel si pentru a ramane doar cu forma pe toata suprafata patratelului, apoi se apeleaza functia **find_shape** pentru a identifica forma.

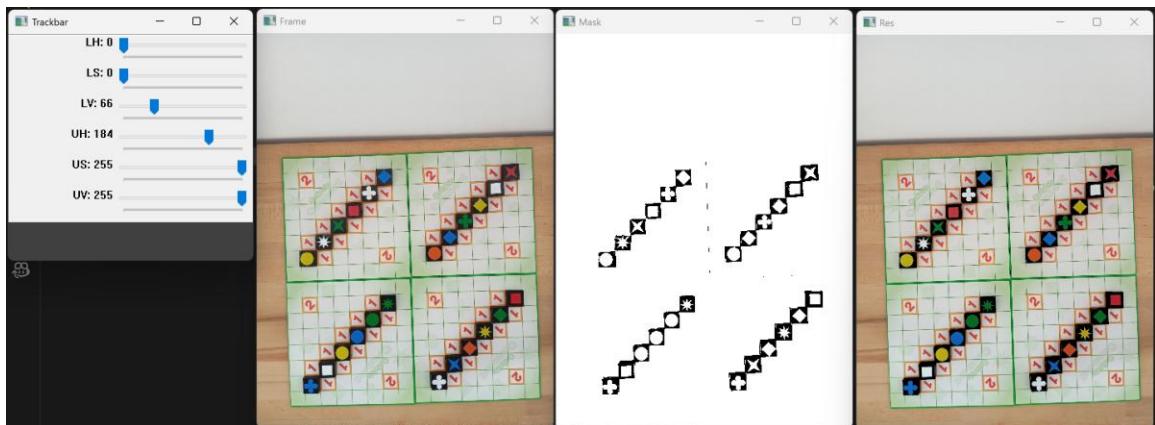


Figura 3

3.1 Detectarea formelor pieselor

La fel ca la **find_digit**, functia **find_shape** foloseste template matching pentru detectarea formei piesei. Se aplica fiecare template cu forma peste patratel si se returneaza numarul formei corespunzatoare lui. Functia returneaza 0 in cazul in care nu se gaseste nicio potrivire cu nicio forma. Daca forma este gasita, se apeleaza functia **find_color** pentru determinarea culorii acesteia.

3.2 Detectarea culorilor pieselor

Functia **find_color** foloseste masuri HSV pentru urmatoarele culori:

- Rosu: lower (177, 146, 0), upper (255, 255, 255)
- Portocaliu: lower (0, 144, 156), upper (24, 255, 255)
- Galben : lower (24, 132, 58), upper (59, 255, 255)
- Verde: lower (64, 206, 31), upper (101, 255, 193)
- Albastru: lower (101, 160, 56), upper (113, 255, 255)

Se aplica fiecare masca peste piesa identificata si returneaza culoarea corespunzatoare. Daca nicio masca nu este potrivita, automat piesa este alba.

Dupa gasirea formei si a culorii piesei din patratel, se retin coordonatele, forma si culoarea piesei in lista pieces. Daca atat functia **find_digit**, cat si **find_shape** returneaza 0, atunci patratelul este gol si nu se face nimic in acest caz.

4. Calcularea scorului

Pentru a calcula scorul, vor fi tratate 3 cazuri:

- Toate piesele adaugate in runda respectiva sunt pe aceeasi linie; acest lucru se verifica folosind functia **have_the_same_line**
- Toate piesele adaugate in runda respectiva sunt pe aceeasi coloana; acest lucru se verifica folosind functia **have_the_same_column**
- In runda respectiva a fost adaugata o singura piesa

In primul caz, se cauta extremitatile din stanga si din dreapta ale liniei folosind functiile **go_left** si **go_right** si se adauga la scor numarul de piese ale acelei linii.

In al doilea caz, se cauta extremitatile de sus si de jos ale coloanei folosind functiile **go_up** si **go_down** si se adauga la scor numarul de piese ale acelei coloane.

In al treilea caz, se verifica daca piesa adaugata are vecini si, in caz afirmativ, se cauta capetele liniei/coloanei pe care o formeaza cu primul vecin gasit si se adauga la scor numarul de piese din linia/coloana respectiva.

Pentru piesele adaugate in runda curenta, se verifica daca sunt pe pozitii cu puncte bonus si, in caz afirmativ, se aduna punctele bonus respective la scor.

5. Scrierea rezultatului in fisier .txt

La final, se apeleaza functia **write_solution** pentru a crea un fisier text cu acelasi nume ca al imaginii curente si se scriu informatiile despre piesele adaugate si scor in el.