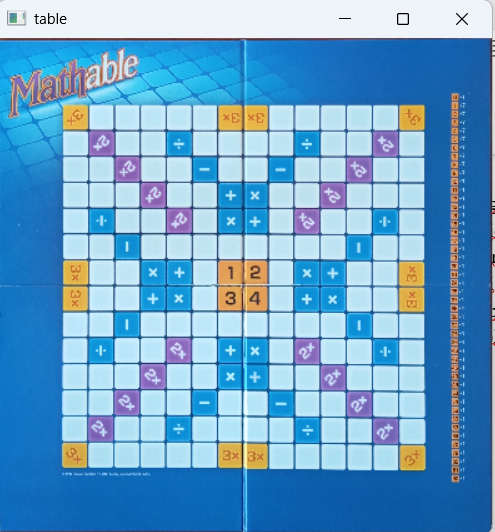
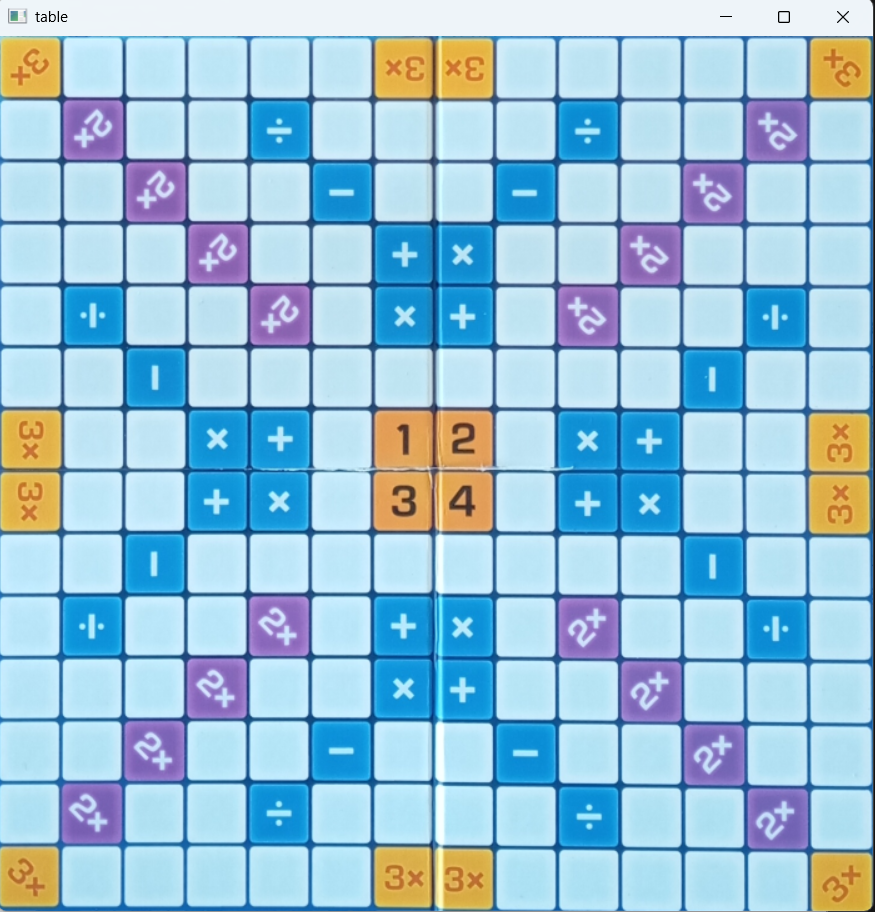
Andronic Smaranda 461

**Proiect 1 CAVA**

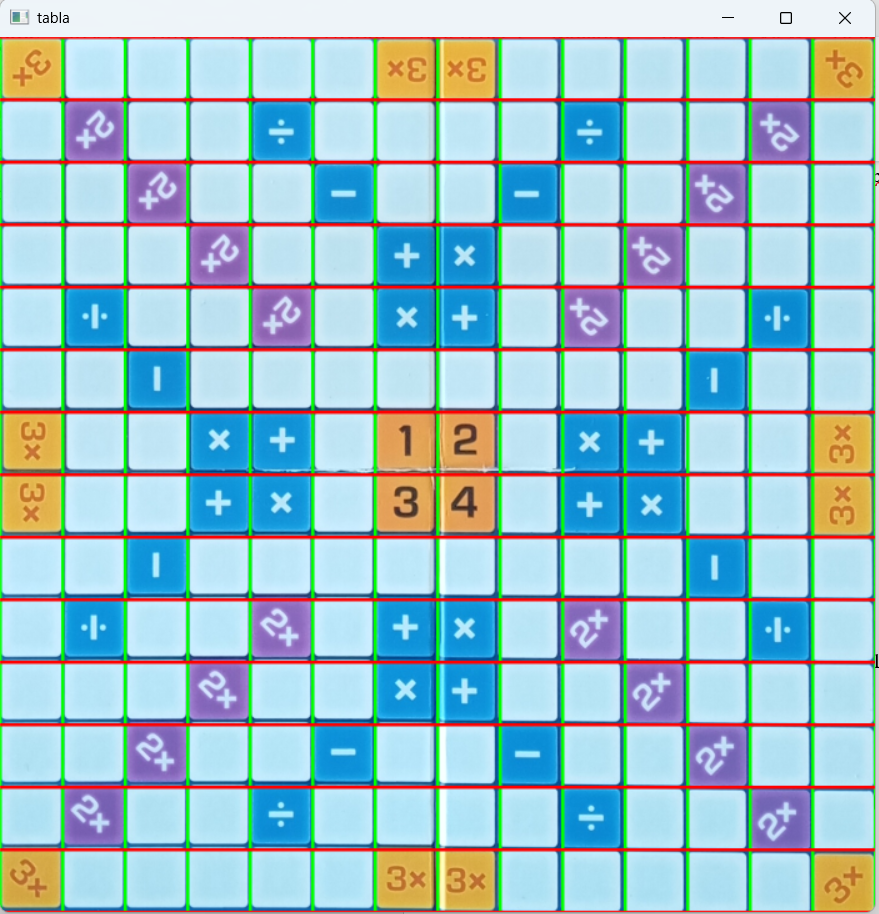
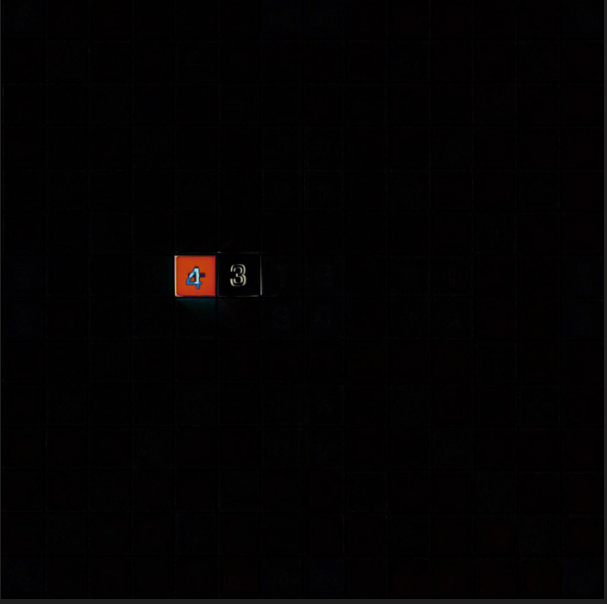
**Task 1 : Identificarea locației piesei**

Iterarea prin imagini se face folosind constant 2 imagini consecutive pentru a ține în evidență ce piesă a fost adăugata de la o mutare la alta. La fiecare joc nou, adică la fiecare 50 de iterații, prima imagine ia valoarea tablei goale. Fiecare imagine este supusă unei preprocesări care constă în eliminarea culorii maro, aplicând o mască, după care se extrage cel mai mare dreptunghi din poză cu funcția boundingRect() din OpenCV, pe baza contururilor obținute folosind funcția findContours() tot din OpenCV. Astfel, obținem întreaga tablă de joc. (Imaginea 1). Întrucât zona de interes a tablei, adică pătratul circumscris 14 x 14, este la dimenisuni predefinite de colțurile tablei deja decupate, putem hardcoda limitele celui de-al doilea pătrat. În urma delimitării extremităților, am folosit funcțiile getPerspectiveTransform() și warpPerspective() pentru a finaliza decuparea și a redimensiona caroiajul la 1400 x 1400 pixeli, astfel încat fiecare pătrat component să aibă 100 x 100 pixeli. Rezultatul final se poate observa în Imaginea 2.

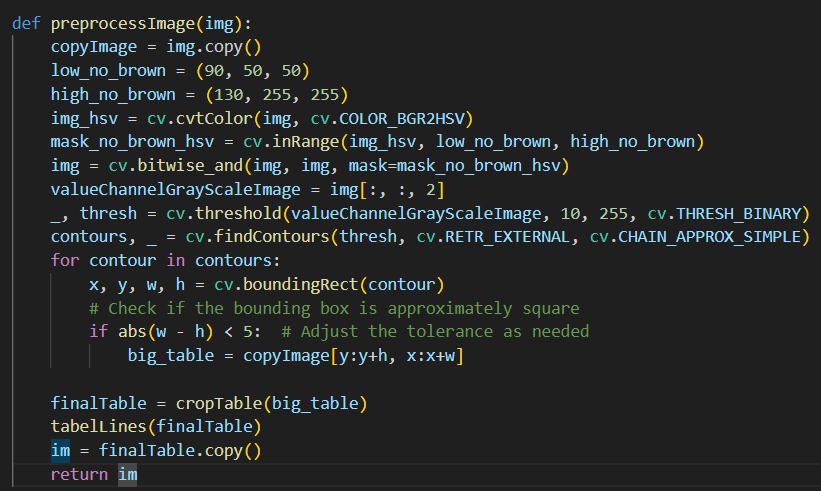
 

Imaginea 1 Imaginea 2

Mai departe am creat delimitările pe linii și coloane pentru tablă, iterând din 100 în 100 de pixeli pe orizontală și veticală. Astfel, am obținut Imaginea 3.

Imaginea 3 Imaginea 4

Codul utilizat pentru a realiza această preprocesare este cel de mai jos   
 

Cel mai important pas din detecția noii piese adăugate pe tablă este folosirea funcției difference() din PIL in această formă :

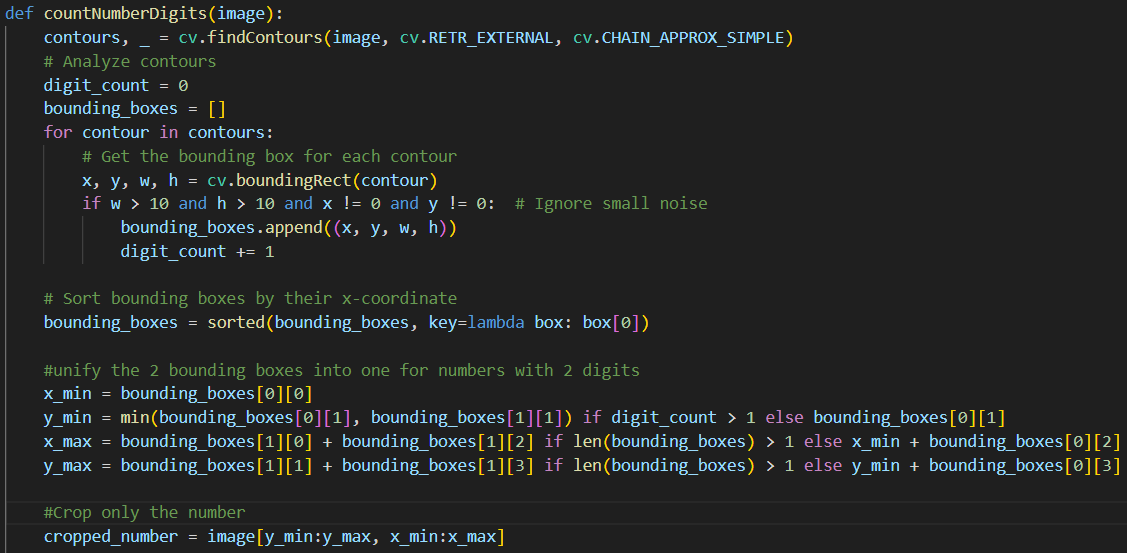
*differenceImage = ImageChops.difference(img1, img2)*

Rezultatul obținut arată ca cel din Imaginea 4. Parcurg această imagine pe linii și coloane din 100 în 100 de pixeli și caut patch-ul de sumă maximă. Acela va corespunde noii piese adăugate. Îi obțin poziția în felul următor :  
 *line = bottom\_right[0] // step*

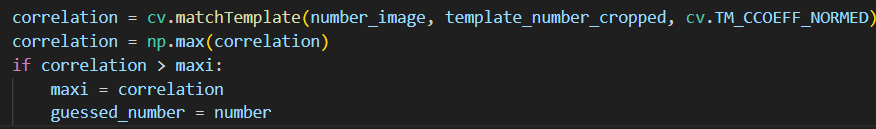
*column = chr(ord('A') + (bottom\_right[1] // step) - 1)*

**Task 2 : Identificarea numărului**

Pentru detectarea numărului de pe piesă am aflat mai întâi dacă numărul nostru este compus din 1 sau 2 cifre. În acest sens, am determinat bounding boxurile aferente pentru acel număr, contorizând numărul lor, ceea ce rezultă în numărul de cifre căutat.



Pasul următor este să transformăm patch-ul care conține numărul nostru într-o imagine binară. Din folderul în care ne-am creat template-uri pentru fiecare număr existent pe tablă (folosindu-ne de imaginea 3 din folderul cu imagini auxiliare), alegem doar numerele al căror număr de cifre corespunde cu patch-ul nostru. Redimensionăm la mărimea patch-ului folosind funcția resize() din OpenCV și binarizăm template-ul nostru, după care aplicăm funcția matchTemplate() din OpenCV. Rezultatul căutat va avea corelația maximă în comparația cu template-ul folosit.



**Task 3 : Calcularea scorului**

Pentru calculul scorului am creat o matrice care respectă constrângerile, bonusurile si piesele centrale 1, 2, 3, 4 deja existente pe tablă. Astfel valorile de -1 reprezintă o căsuță goală, iar restul valorilor sunt metadate despre tablă.

A screenshot of a computer program

Description automatically generated

Mai întâi verificăm dacă piesa se află pe o casetă cu o constrângere de operație. După care, pentru fiecare 2 vecini din stânga, dreapta, sus și jos, dacă este cazul, verificăm dacă se respectă contrângerea existentă sau, dacă nu, orice operație validă. În secvența de mai jos verific pentru vecinii din stânga condițiile algebrice.

A screen shot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generated

La final, scorul devine scorul vechi înmulțit cu numărul de operații satisfăcute și înmulțit cu bonusul aferent. Toate aceste calcule asupra scorului se desfășoară atâta timp cât suntem pe tura unui anumit jucător, care operează într-un interval de mutări, definit în cod între 2 capete left și right. Dacă suntem la o mutare care depășește acest interval, însemnă că s-a intrat pe tura celuilat jucător și restartăm scorul și intervalul.

A computer screen shot of a program code

Description automatically generated