Documentație Tema 1 CAVA

Potlog Ioana

December 3, 2024

1 Task-ul 1

1.1 Extragerea tablei de joc

Funcția extract_table() are rolul de a decupa tabla de joc dintr-o imagine, pregătind-o pentru procesare. Imaginea este convertită în spațiul de culoare HSV, iar apoi se creează o mască care izolează pixelii albastri într-un interval specificat. Apoi, se detectează marginile folosind algoritmul Canny și se identifică contururile exterioare. Pentru fiecare contur, se determină colțurile (stânga sus, dreapta sus, stânga jos, dreapta jos), iar conturul cu cea mai mare arie este considerat conturul tablei de joc. Dacă sunt identificate toate cele patru colțuri, se aplică o transformare de perspectivă pentru a "așeza" tabla de joc, redimensionând-o la o dimensiune fixă (940x940 pixeli). Funcția returnează imaginea prelucrată, care conține doar tabla de joc extrasă.

După ce am extras tabla de joc, pentru a rămâne doar cu careul am folosit funcția crop_board(), care primește o imagine și coordonatele celor patru colțuri ale regiunii de interes (hardcodate în variabila corners) și transformă acea regiune într-o vedere plană, de dimensiuni fixe (1400x1400 pixeli), am redimensionat pentru claritate.

De asemenea, am calculat coordonatele liniilor verticale si orizontale pentru a delimita celulele tablei de joc într-o grilă de 14x14 celule. Liniile verticale sunt calculate prin fixarea coordonatei x și varierea coordonatei y de la 0 la 1399, în pași de 100 pixeli. Similar, liniile orizontale sunt determinate prin fixarea coordonatei y și varierea coordonatei x.

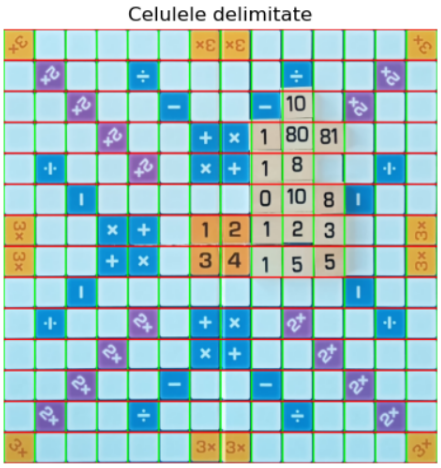

Figure 1: Poza cu tabla de joc cu celulele delimitate

1.2 Detectarea locatiei piesei adaugate

Funcția detect_added_piece() identifică celula de pe tabla de joc în care a fost adăugată o piesă nouă, comparând imaginea actuală cu imaginea precedentă, imaginile rămân color. Tabla este împărțită într-o grilă de 14x14 celule, fiecare celulă având o dimensiune calculată pe baza dimensiunii tablei (board-size) și a lățimii liniilor de grilă (line-width).

- cell-size = (board-size - rows * line-width) // rows

Funcția parcurge toate celulele pe rând și compară secvențial celulele corespunzătoare din cele două imagini, utilizând suma diferențelor absolute între pixeli pentru a detecta modificările.

- diff-sum = np.sum(np.abs(initial-cell.astype(int) -current-cell.astype(int)))

Celula cu cea mai mare diferență este considerată locația piesei adăugate, iar funcția returnează coordonatele acesteia (rând și coloană).

2 Task-ul 2

2.1 Crearea template-urilor

Am extras celulele din imaginea auxiliară numărul 3 cu ajutorul functiei extract-cells(). Celulele sunt identificate pe baza coordonatelor liniilor de grilă, iar marginile albastre (-5 sau +5) dintre celule sunt eliminate pentru a izola piesele.

Fiecare celulă este convertită în tonuri de gri (cv.cvtColor) și binarizată (cv.threshold). Marginile negre inutile sunt eliminate cu funcția remove-margins(). Imaginea rezultată este estompată (cv.blur) și binarizată adaptiv pentru a accentua detaliile relevante. Contururile sunt detectate cu cv.findContours(), iar contururile care depășesc o anumită arie (MIN-CONTOUR-AREA) sunt analizate. Dacă dimensiunile conturului (înălțime și lățime) se încadrează în limitele definite (MIN-HEIGHT și MIN-WIDTH), piesa este decupată. Am făcut această verificare, deoarece codul identifica doar o cifră, în loc să identifice tot numărul.

De asemenea, pentru a rezolva problema în care funcția cv.matchTemplate() îmi detecta doar o cifră în loc să detecteze tot numărul, am adăugat câteva linii albe pe partea stângă și pe partea dreaptă a cifrelor.

Figure 2: Template-ul cifrei 4, pus pe un fundal negru pentru a se vedea marginea lățită

2.2 Funcția care identifică numărul reprezentat pe piesă

Funcția number_classification() are rolul de a clasifica o celulă (patch) ca fiind unul dintre numerele pentru care există șabloane salvate. Aceasta parcurge lista de template-uri (templates/j.jpg) și utilizează metoda cv.matchTemplate(), pentru a calcula corelația normalizată între patch-ul dat și fiecare șablon. Corelația maximă este determinată, iar numărul corespunzător șablonului cu cea mai mare similaritate este returnat, împreună cu șablonul.

- cv.matchTemplate(patch, img_template, cv.TM_CCOEFF_NORMED)

3 Task-ul 3

3.1 Implementarea tablei și detectarea celulei adăugate

Funcția define_matrix() inițializează tabla de joc, marcând celulele cu proprietăți distincte (”2x”, ”3x”, ”+”, ”-”, ”*”, ”:”), iar pe restul cu ”empty”.

În timpul fiecărei mutări, se analizează imaginile înainte și după plasarea piesei pentru a identifica coordonatele celulei modificate, folosind funcțiile extract_table(), crop_board() și detect_added_piece(). Apoi, celula detectată este prelucrată pentru a determina valoarea numerică a piesei plasate folosind funcția number_classification(). Aceste funcții au fost descrise și mai sus.

3.2 Calcularea scorului

Odată plasată piesa, algoritmul află calculele posibile pe baza celulelor vecine. Pentru fiecare direcție (dreapta, stânga, sus, jos), se evaluează expresiile matematice folosind valorile numerice ale pieselor vecine și operatorul specific celulei curente, dacă este cazul. Dacă rezultatul calculului coincide cu valoarea piesei plasate, punctele se adaugă la scorul jucătorului. Celulele speciale, cum ar fi cele marcate cu 2x sau 3x, multiplică scorul aferent piesei în funcție de proprietatea celulei.

References

M-am ajutat de o bucățică de cod de pe site-ul Stack Overflow, pe care am folosit-o la extragerea numerelor din piese pentru template-uri.