Concepte si Aplicatii in Vederea Artificiala

Tema 1 - Calculator automat de scor pentru Scrabble

- Task 1:

Pentru realizarea primului task am folosit: o matrice pentru a retine pozitiile pieselor de la o mutare la alta, functia de extragere a tablei de joc, functia de extragere a pozitiilor tuturor pieselor si functia de gasire a pieselor mutate doar in jocul respectiv.

```
img_cropata = extrage_tabla(img)
```

Aceasta functie returneaza tabla de joc decupata si rotita. Primul lucru pentru a ajunge la acest rezultat a fost de a aplica filtre morfologice pe imaginea tranformata din BGR in HSV. Acest fapt a pune in prim plan muchiile ce ne erau de interes. Dupa aplicarea filtreror am scos muchiile din care au rezultat contururile dorite. Dupa gasirea contururilor, cu ajutorul functiei *detectare_colturi(contours),* am gasit colturile tablei. Aceasta verifica fiecare contur gasit si il valideaza folosid o functie de arie. Cum setul de date contine poze de intensitati de lumina diferita, in aceasta functie saturatia minima se schimba in cazul in care nu este gasita tabla. Dupa aceea, tabla este decupata folosind coordonatele acesteia.

```
pozitii_piese(img_cropata)
```

Aceasta functie returneaza o matrice reprezentand pozitiile tuturor pieselor la mutarea respectiva. Pe aceasta am aplicat filtre morfologice pentru a scapa de elementele nefolositoare din fundal si un threshold pentru a evidentia si tranforma intr-o imagine binara piesele de pe masa. Apoi, stiind marimea tablei si numarul de patrate pe verticala, respectiv orizontala, am putut decupa fiecare patrat si sa verificam daca acesta contine sau nu o piesa in functie de threshold ul acestuia.

Avand o matrice cu pozitiile mutarii precedente, si o matrice cu pozitiile mutarii actuale, am creat functia *pozitii_mutari,* care compara cele 2 matrici, si returneaza un vector cu pozitiile noilor piese.

- Task 2:

Pentru a realiza acest task m am folosit de tabla decupata si vectorul de mutari gasit la primul task, iar cu ajutorul aplicatiei Photos din Windows, am decupat fiecare piesa pentru a fi folosite ca template. Acestea au o dimensiune de 120x130px. Am folosit functia din cv2, matchTemplate, pentru a detecta ce piesa se afla pe un patrat, dar inainte de astea am aplicat filtre morfologice atat pe patch, cat si pe piesa. De asemenea, aceasta verificare se realizeaza pe piesa rotita din 2 in 2 grade la stanga si la dreapta pentru a ne asigura ca piesa pusa pe tabla nu prezinta o diferenta de pozitie fata de cea din template. Template ul care se potriveste cel mai bine este retinut si in aceeasi maniera ca la gasirea pozitiilor pieselor este returnat un vector cu literele gasite.

Task 3:

Pentru a calcula scorul am creeat o matrice de litere, ce reprezinta literele pieselor de pe tabla, vectorul cu pozitiile ultimei mutari si literele mutarii. Aceasta matrice este actualizata folosind functia actualizare_matrice_litere. Primul lucru pe care il fac pentru a calcula scorul este sa pun in doi vectori: linii si coloane, coordonatele pieselor puse in mutarea respectiva. Daca lungimea vectorului de linii este egala cu 1, inseamna ca piesele au fost puse pe orizontala. Inainte de a calcula scorul acestuia, trebuie sa extindem capetele vectorului pentru a include si piesele care nu apartin mutarii. Dupa ce facem acest lucru apelam functia calculeaza_scor_orizontala, ce calculeaza scorul cuvantului pus pe orizontala si verifica, urmat sa fie calculat, daca contine si cuvinte puse pe verticala. Asemanator este tratat si cazul in care lungimea vectorului de coloane este egal cu 1. Mai este de tratat si cazul in care ambele sunt egale cu 1, iar in acest caz extind acesti vectori, dupa care putem deduce cum au fost formate cuvintele.