

Proiect 1 CAVA

Turcu Virgil

In acest proiect am creat un script de python ce ia ca input imagini ce contin o tabla de Scrabble si calculeaza ce piese au fost adaugate si care este punctajul lor. In arhiva trimisa se afla datelele urmatoare :

- un fisier readme cu instructiuni despre cum trebuie sa fie rulat scriptul si ce librarii sunt folosite in script.
- Un fisier .py ce contine tot scriptul ce trebuie rulat.
- Un folder ce contine 20 de imagini cu template-uri pentru piesele jocului de scrabble. Aceste template-uri au fost alese special pentru a face scriptul sa fie cat mai eficient. Folosirea altor template-uri poate duce la o performanta mai slaba a programului.
- Aceasta documentatie ce va explica pe scurt cum functioneaza programul.

Extragerea careului

Primul pas al programului este extragerea careului. In acest pas mi-am propus sa extrag doar careul din fiecare poza. Pentru inceput, am observat ca o parte din poza reprezenta doar masa pe care statea tabla, asa ca am cropat poza pentru a elimina cat mai mult din masa. Al doilea pas a fost sa fac o masca HSV pentru poza ce va evidentia contururile din imaginea originala. Cea mai potrivita masca am dedus-o printr-un proces trial and error ajutandu-ma si de functia filtrarea_culorilor din laboratorul 7. In acest caz am realizat ca o masca HSV cu dimensiunile ((0, 255), (0, 140), (0, 255)) este cea mai potrivita. Pe aceasta masca HSV am aplicat functia de findContours din openCV pentru a gasi toate contururile. Am parcurs toate contururile gasite de lungime mai mare ca 3 (pentru a evita exceptiile) si am gasit cele 4 colturi ale careului. Am luat imaginea delimitata de cele 4 colturi si am transformat-o intr-o imagine de marime 600x600 folosind functia getPerspectiveTransform din openCV.

Determinarea pozitiilor ocupate de piese

Al doilea pas al programului este determinarea pozitiilor care sunt ocupate de piese. Pentru acest pas am folosit o masca HSV pentru a gasi pozitiile din care ce sunt mai luminate decat celelalte si am calculat media acestui patch fata de o variabila determinata prin trial and error. In cazul de fata daca media patch-ului era mai mare strict ca 8 inseamna ca acolo se afla o piesa.

Memorarea template-urilor

Al treilea pas al programului este determinarea template-urilor. In acest pas am folosit o functie memoreaza_templates ce se uita in cele doua imagini cu piesele jocului din folder-ul imagini_auxiliare, si salveaza fiecare patch, cropuit cu 4 pixeli in fiecare directie, intr-un folder numit templates. Dupa ce toate 20 de patchuri au fost salvate in folder-ul templates, le-am redenumit manual pe fiecare pentru a ave aca titlu litera ce se afla in patch-ul respectiv. Ulterior functia memoreaza_templates a fost modificata pentru a putea prelua toate template-urile din toate jocurile. Aceasta a fost o incercare nereusita de

a mari acuratetea programului. Avand in vedere ca templates-urile fusesera deja salvate, am decis ca functia memoreaza_templates nu va mai fi folosita deci nu am mai modificat-o la starea initiala. In programul din arhiva, functia memoreaza_templates memoreaza pentru fiecare runda din joc toate piesele care apar pe tabla si le salveaza intr-un fisier ce are ca titlu litera din imaginea respectiva, urmata de niste numere pentru a ne asigura ca fisierele au nume diferite.

Clasificarea unei litere

Al patrulea pas a fost sa atribui unui patch cu o piesa pe el, o litera ce se potriveste literei de pe piesa. Pentru acest lucru am creat o functie clasifica_litera ce ia ca input un patch si path-ul catre folderul ce contine templates-urile. Functia preia patch-ul si il compara cu fiecare template din folder-ul templates folosind functia matchTemplate din openCV. Gradul de asemanare maxim si litera corespondenta sunt retinute si la final este returnata litera cu gradul de asemanare cel mai mare.

Calcularea punctajului pe fiecare runda

Ultimul pas este calcularea punctajului pentru fiecare runda a jocului. Pentru acest pas, am inceput prin a hardcoda o matrice de 15x15 unde pe fiecare pozitie se afla un numar ce corespunde cu tipul de patrat de pe acea pozitie. In cazul de fata :

- 1 – patrat galben
- 2 – patrat bleu
- 3 – patrat bleumarin
- 4 – patrat roz
- 5 – patrat grenă

De asemenea, pentru a respecta adnotarile din fisierul solutie a trebui sa hardcodez un dictionar unde fiecare poz de la 1 la 15 reprezinta litera corespondenta din alfabet. 1 – A , 2 – B etc. Un alt dictionar hardcodat este cel care retine punctajul pentru fiecare litera. In cazul acesta cheia din dictionar este litera iar valoarea este punctajul pe care il reprezinta. Urmatorul pas este calcularea scorului propriu zis. Algoritmul pe care l-am ales functioneaza in felul urmator. La fiecare runda se adauga un cuvant nou pe verticala sau orizontala. Eu gasesc prima piesa noua adaugata folosind o matrice anterioara, ce retine configuratia tabliei inainte de runda curenta si o matrice curenta ce retine configuratia curenta a tabliei. Dupa ce am gasit prima piesa noua, parcurg in sus pana nu mai gasesc piese, si apoi incep sa cobor uitandu-ma daca piesa curenta este nou adaugata. Daca gasesc 2 sau mai multe piese nou adaugate asta inseamna ca noul cuvant este pe verticala. Ma opresc in momentul in care nu mai am piese unde sa cobor.

Acest rationament este aplicat si pentru a verifica daca cuvantul este pe orizontala, doar ca in loc sa parcurgem de sus in jos, parcurgem de la dreapta la stanga. De asemenea la ambele parcurgeri calculam scorul cuvantului. La final daca cuvantul este pe orizontala adaugam scorul cuvantului de pe orizontala la scorul final iar daca cuvantul este pe verticala adaugam scorul cuvantului de pe verticala la scorul final. Dupa ce am parcurs cuvantul pe orizontala si pe verticala incepand de la prima litera adaugata, luam urmatoarele litere adaugate si le parcurgem in felul urmator:

Daca noul cuvant este pe verticala, atunci luam fiecare litera adaugata, in afara de prima, si le parcurgem pe orizontala, pentru a gasi daca si ele creaza cuvinte noi pe orizontala, adaugand scorul acestor cuvinte noi pe orizontala la scorul final.

Daca noul cuvant este pe orizontala, atunci luam fiecare litera adaugata, in afara de prima, si le parcurgem pe verticala, pentru a gasi daca si ele creaza cuvinte noi pe verticala, adaugand scorul acestor cuvinte noi pe verticala la scorul final.

De asemenea verificam daca la o runda au fost adaugate 7 litere pe tabla, in acest caz, runda respectiva primeste 50 de puncte extra. Dupa ce am pastrat intr-o lista pozitiile pieselor noi aparute, litere ce se afla pe piese si scorul rundei , aceasta lista este pusa ca element intr-o lista mai mare ce pastrea toate rundele jocului curent. Dupa ce se termina jocul curent, toate rundele sunt scrise in fisierul solutie corespondent rundei.