

Rešavanje mica igre na osnovu učitane fotografije table

Jovana Jovanović (SW26/2015), Jovo Šunjka (SW17/2015)

Fakultet tehničkih nauka, Novi Sad

Problem

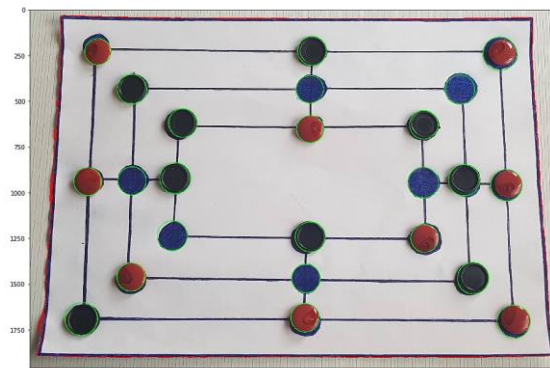
Mnogi ljudi igraju micu, ali često ne znaju koji potez da odigraju, a da bude najbolji za njih. Ovim rešenjem mi želimo da im pomognemo, tako što će moći da slikaju njihovu trenutnu tablu za igru, i da odeberu jedan od tri algoritma (Minimax, Reinforcement learning, Deep RL), koji će im savetovati koji potez mogu da odigraju sledeći.

Šta je mica?

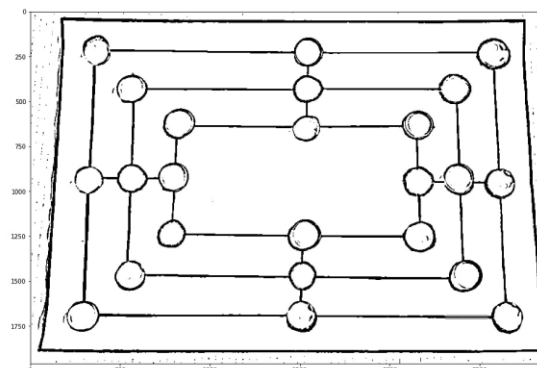


Mica je igra koja se igra na specijalno napravljenoj tabli. Igraju je 2 igrača, koji imaju po 9 figura. Sastoji se od 3 faze, a to su faza postavljanja figura, faza pomeranja figura i faza skakanja. Tabla se sastoji od 24 polja na koja mogu da se postave figure. Igrači igraju naizmenično jedan pa drugi, sve dok ne sastave micu, odnosno 3 sopstvene figure spojene vertikalno ili horizontalno. Kada igrač sklopi micu dozvoljeno mu je da pojede jednu od protivničkih figura. Igra se završava kada jedan od igrača ostane sa 2 figure ili kada više ne može da pomeri nijednu figuru.

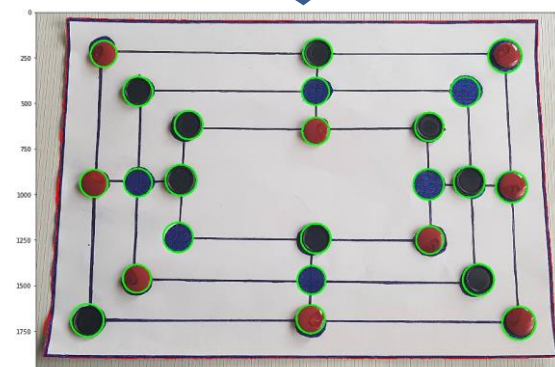
Procesiranje slike



Slika 1. ulazna slika



Slika 2. Binarna slika



Slika 3. Izdvojena tabla i figure

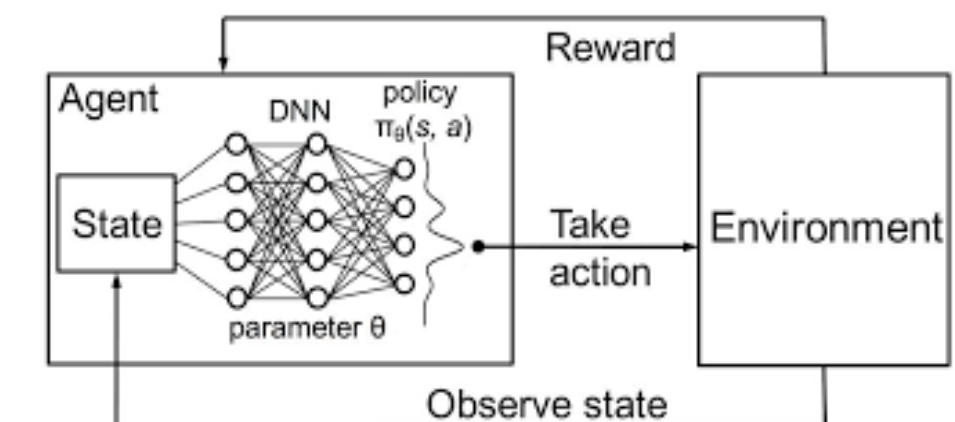
Result:

1, 1, 2, 1, 1, 1, 1, 2;
2, 0, 1, 0, 2, 2, 0, 0;
2, 2, 0, 2, 1, 0, 2, 1 |
0, 0

Proces obrade slike započinjemo učitavanjem slike za obradu, zatim učitanu sliku pretvaramo u grayscale sliku, a onda sliku zamutimo uz pomoć medianBlur funkcije, kako bismo smanjili detektovanje šumova. Koristeći adaptiveThreshold funkciju, sliku pretvaramo u binarnu (slika 2). Nakon što smo dobili binarnu sliku tražimo sve konture na slici, pomoću funkcije findContours i izdajemo tablu sa slike. Posle izdvajanja table, pomoću Hougha pronalazimo krugove koji se nalaze unutar table i određujemo njihove pozicije. Pozicije krugova određujemo tako što krenemo od krugova koji se nalaze najbliže gornjem levom i donjem desnom ćošku, a zatim nađemo njima najbliže krugove i sve tako dok ne pronađemo pozicije svih 24 kruga. Kada pronađemo sve krugove, ostaje nam još samo da odredimo kom igraču pripadaju, to radimo na osnovu boje, tako što uzmemo vrednost centralnog piksela kruga i odredimo koliko figura nije postavljeno na tabli. I na kraju, kao rezultat vratimo stanje table kao string. 1 i 2 predstavljaju različite igrače, a 0 slobodno polje, ";" novi nivo na tabli i na kraju posle "|" stoji broj figura koje nisu postavljene.

Deep reinforcement learning

Deep reinforcement learning je oblast mašinskog učenja fokusirana na obučavanje agenta koristeći neuronsku mrežu i tehniku experience replay. Kako radimo akcije, ishode (S_t, A_t, R_t, S_{t+1}) čuvamo u memoriji (replay memory) – ovo je ustvari skup našeg iskustva (experience). Neuronsku mrežu obučavamo na mini-podskupovima odabranim na slučajan način iz replay memorije. Pošto biramo na slučajan način, mreža neće učiti iz niza uzastopnih akcija. Učenje na osnovu niza uzastopnih semplova se empirijski pokazalo kao problematično.



Na početku treniranja što više igramo random poteze, kako bi bismo što više istražili okruženje. Kasnije se sve više oslanjamo na poteze koje nam preporuči obučena neuronska mreža.

Analiza

- Uspešnost procesiranja slike je detektovano upoređivanjem početne slike i dobijenog rezultata
- Uspešnost Deep reinforcement learninga je posmatrana kroz ostvarene poraze i pobeđe nad ranije napravljenim algoritimima (Minimax, Reinforcement learning)