

Prepoznavanje i rešavanje rubikove kocke

Dalibor Malić, Darko Tica, Zoran Jankov

Fakultet Tehničkih Nauka, Novi Sad

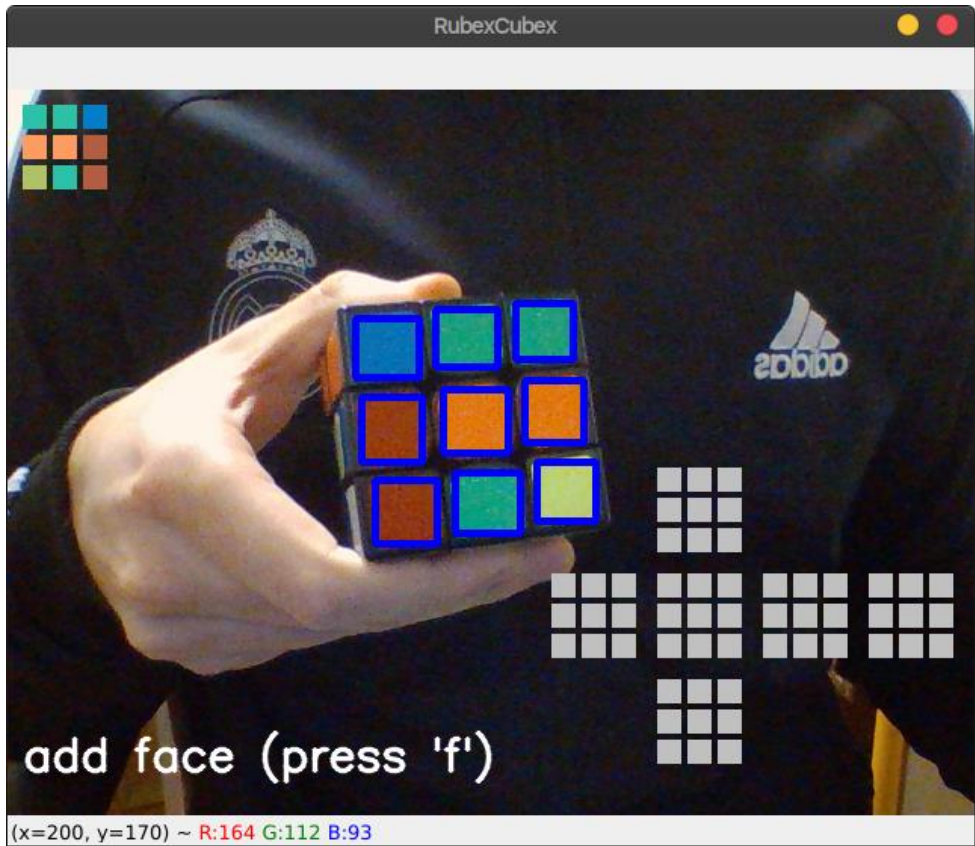
Uvod

Još od samog kreiranja Rubikove kocke, problem njenog rešavanja, odnosno potraga za optimalnim algoritmom za njeno slaganje predstavlja jedan od bitnih programerskih i matematičkih problema. Iako je došlo do značajnih pomaka u vidu pronalaženja algoritama kao i matematičkih dokaza (maksimalan broj poteza za rešavanje 20 – “**Božiji broj**”), tačan algoritam koji optimalnim putem vodi do rešenog stanja još uvek nije poznat. Stoga, jedan od ciljeva primene veštačke inteligencije u njenom rešavanju jeste pokušaj pronalaska upravo takvog algoritma.

Prepoznavanje kocke

Prepoznavanje kocke se vrši tako što je potrebno da korisnik snimi svaku pojedinačnu stranu kocke, prilikom čega se prepoznaju obojeni kvadrati u okviru svake njene strane. Prvi korak prilikom detekcije jeste pretvaranje prosleđene slike u HSV format. Na taj način, vrši se lakša analiza i procesiranje slike. Nakon ove konverzije, primeni se **Gaussian blur** nad value slojem slike (zbog otklanjanja šuma). Zatim se primenjuje **Canny edge detection** metoda, radi pronalaženja ivica unutar slike, pri čemu su njeni parametri utvrđeni empirijski. Na osnovu utvrđenih ivica pronalaze se konture unutar slike (zatvoreni oblici). S obzirom da su obojene strane Rubikove kocke u obliku kvadrata, potrebno je izdvojiti samo konture u tom obliku. Ovo filtriranje se vrši proveravanjem broja stranica svake konture (mora ih imati 4), kao i izračunavanjem i poređenjem površine konture sa površinom kvadrata opisanim oko nje (pri čemu mora doći do poklapanja njihovih površina sa određenom tolerancijom). Takođe, proverava se da li svaka konturaima odgovarajući broj suseda (svaki kvadrat kocke ima susede, čime uklanjamo konture sa slike koje su izdvojene i predstavljaju šum). Nakon što su prepoznata polja sa prikazane strane kocke, potrebno je odrediti boju svakog od njih. To se vrši tako

što se iz onog dela originalne slike gde se nalazi određeno polje traži dominantna boja unutar njega.



Detekcija kocke

Rešavanje rubikove kocke

Kako je Rubikova kocka puzzle koja se sastoji od velikog broja mogućih stanja (43 triliona), obična pretraga u najvećem broju slučajeva ne bi bila moguća zbog ograničenosti resursa i vremena. Usled toga, potrebno je razviti heurističku funkciju koja će svakom stanju dodeliti numeričku vrednost koja opisuje njegovu udaljenost od cilja. Na ovaj način, prilikom rešavanja pretraživaće se samo stanja koja vode direktno ka cilju, čime će se značajno ubrzati proces pretrage. Glavna ideja rešavanja primenjena u ovom radu jeste korišćenje dubokog pojačanog učenja u cilju pronalaska heurističke funkcije, a zatim i njena primena u okviru **greedy search** algoritma radi pretrage narednog poteza.

Duboko pojačano učenje

Duboko pojačano učenje se zasniva na korišćenju neuronske mreže koja pored ulaznog i izlaznog sloja sadrži četiri skrivena sloja. Ulaz u mrežu predstavlja niz od 288 brojeva (koji imaju vrednosti 0 ili 1) koji predstav stanje kocke. Ovaj niz se dobija tako što se svako polje kocke pretvori u niz dužine šest brojeva uz pomoć **one-hot encoding** metode, što je urađeno radi lakšeg i bržeg treniranja. Svaki sloj unutar mreže koristi **ReLU** funkciju aktivacije, i na izlazu iz svakog unutrašnjeg sloja je primenjena **batch** normalizacija. Kao izlaz mreže dobija se nenegativna vrednost koja predstavlja ocenu udaljenosti prosleđenog stanja kocke do rešenja.

Treniranje

Kao algoritam za pojačano učenje (**reinforcement learning**) odabran je **approximate value iteration** metod, koji se zasniva na iterativnom učenju koraka rešavanja kocke. Vrednost kocke u stanju k (koje je udaljeno k koraka od cilja) se evaluira kao heuristika stanja kocke u stanju k-1 uz dodavanje cene poteza koja je 1 za svaki potez. Na ovaj način, ciljno stanje je označeno vrednošću 0, dok se svako naredno računa kao vrednost prethodnog uvećano za 1. Treninom mreže na stanju koje je jedan potez daleko od ciljnog, a zatim stanju udaljenom dva poteza od cilja itd., dolazi do njenog učenja kroz iteracije.

Iako je glavna ideja rada bila treniranje neuronske mreže koristeći prethodno pomenuti algoritam, treniranje mreže isključivo na taj način se pokazalo nemogućim. Jedan od glavnih razloga jeste nedostatak fizičkih resursa za treniranje (nedovoljna RAM memorija i snaga grafičkih kartica). Stoga, u ovom radu je mreža prvo istrenirana nadgledanim učenjem, pri čemu su se kao trening podaci koristile pozicije generisane nasumičnim potezima od

ciljnog stanja, dok su se kao rezultati koristili brojevi poteza koji su doveli do tog rešenja. Na taj način, mreža je inicijalno naučena da procenjuje udaljenost svakog stanja od cilja, a zatim je uz pomoć pomenutog metoda pojačanog učenja mreža dodatno istrenirana čime je postignut značajan napredak u performansama. Nakon što je neuronska mreža istrenirana, sama pretraga rešenja kocke se vrši uz pomoć **greedy search** metode, koja kao sledeći potez kocke bira potez koji vodi do stanja koje ima najmanju heuristiku, odnosno do stanja koje je najbliže ciljnom stanju kocke.

Rezultati

Kod detekcije kocke, u normalnim svetlosnim uslovima, program uspeva da detektuje boje na strani kocke iz prvog pokušaja u 96% slučajeva. Podešavanjem svetla, ili pomeranjem kocke ispred kamere, preciznost detekcije u narednim potezima raste, i dostiže 99%.

Kod rešavanja kocke, s obzirom na ograničene resurse prilikom treniranja, program uspeva da nađe rešenje do 11 poteza udaljenost od ciljnog u roku od 30s u 94% slučajeva. Na 12 poteza od cilja, program uspe da reši kocku u 90% slučajeva u roku od 30s. Nakon toga, sa sve većim brojem poteza od cilja, rešivost kocke brzo opada, te ti slučajevi nisu analizirani.

Potezi	Supervised 1 *samo nadgledano	Supervised 2 *dotrenirano nadgledano	Supervised 2 *dotrenirano RL
9	89%	100%	100%
10	86%	88%	100%
11	60%	74%	94%
12	32%	60%	90%

Potezi – broj poteza do krajnjeg stanja