Poročilo RIS

Boštjan Bohte, Beno Šircelj, Boris Karamatić

Ljubljana University, Večna pot 113, Slovenia,

Abstract. Keywords:

1 Metode

1.1 Prepoznavanje znakov in obrazov

Prepoznavanje znakov in obrazov je sestavljeno iz dveh delov, klasifikacijski model in ustvarjanje učne množice. Slednji se uporablja samo za pripravo klasifikacijskega modela in je nato neaktiven čez celotni proces, drugi del pa omogoča razpoznavanje različnih znakov ali obrazov in je vedno aktiven. Ker je teren zaprt prostor in so skoraj idealni pogoji, ni bilo potrebno narediti velike učne množice.

Ustvarjanje učne množice znakov: Prijavimo se na temo 'traffic_signs', ki je bil implementiran s strani predmeta in iz njega pridobivamo podatke o detektiranem znaku. Zanima nas samo velikost in pozicija znaka na sliki. Iz slike izluščimo znak in ga nastavimo na dolžine 50x50 pikslov. Znaku nato izenačimo histogram za vsak barvni kanal R,G,B, da dobi boljšo odpornost na spremembe svetlosti v prostoru. Znak nato shranimo z oznako v posebno mapo, kjer bo služil za učno množico modela. Naredili smo približno 30 učnih primerov za vsak znak, ker se je izkazalo, da je to zadostno število in je s tem bilo tudi učenje krajše.

Prepoznavanje znakov: Pri prepoznavanju znakov ustvarimo najprej klasifikacijski model. Za to nalogo smo določili, da bo bil to model k najbljižjih sosedov, ker je prosto dostopen v knjižnjici OpenCV. Iz mape kjer so označene slike znakov, preberemo celotno sliko kot matriko, jo sploščimo v vektor in podamo modelu kot učni primer. Učenje je hitro, saj rabi malo učnih primerov zato se to naredi vsakič, ob zagonu procesa.

Prijavimo se na temo 'traffic_signs'(detektor znakov), kjer nas zanima pozicija in velikost znaka. Enako kot pri učenju izrežemo ven znak, ga nastavimo na dolžine 50x50 pikslov in na vseh barvnih kanalih RGB izenačimo histogram. Matriko spremenimo v vektor in podamo na vhod modela. Model nam vrne razdaljo in napovedan znak. Razdalja lahko služi kot neko verjetje, večja kot je, manjša je zanesljivost te napovedi. Da proces vrne kateri znak je prepoznal, mora bit razdalja manjša od nastavljenega praga in znak mora biti v kratkem intervalu časa zaznan vsaj petkrat. Detektor pošilja detektirane znake okoli 10x na sekundo, kar pomeni, da bo detektiran znak prepoznal v približno 0.5 sekundah. Čas je ključnega pomena, saj bi lahko v daljšem zamiku zgrešil znak. Za večjo robustnost se zato bolj zanašamo na prag razdalje.

Ustvarjanje učne množice obrazov Prijavimo se na temo 'faces', ki je bil implementiran s strani predmeta in iz njega pridobivamo podatke o detektiranem obrazu. Zanima nas lokacija in velikost obraza na sliki. Iz slike izrežemo detektiran obraz in spremenimo v sivinsko sliko. Sliki nato še izenačimo histogram, da je bolj odporna na spremembe svetlobe. Izrezek slike nato shranimo pod oznako obraza, ki bo služila kot učni primer. Za vsako obraz smo naredili približno 20 učnih primerov.

Prepoznavanje obrazov Za prepoznavanje obraza uporabimo model LBPH, ki je implementiran v OpenCV knjižnjici in je prosto dostopen. Za vhod rabi sivinske slike obrazov, velikost matrike pa ni potrebno da je enaka, zato je učenje modela enostavno. Samo preberemo iz mape vse shranjene slike obrazov in jih uporabimo kot učno množico. Zaradi majhne učne množice tudi to učenje traja kratek čas, zato se učenje izvaja ob vsakem zagonu procesa.

Prijavimo se na temo 'faces' (detektor obrazov) in nas zanima pozicija in velikost detektiranega obraza. Iz teh podatkov izrežemo obraz iz slike, izrezek pretvorimo v sivinsko sliko in izenačimo histogram. Obdelano sliko vstavimo v naučeni model in nam vrne napoved in verjetnost te napovedi. Večja je verjetnost, bolj zanesliva je napoved. Tudi tokrat uporabimo prag za verjetnost in število detekcij v danem intervalu. Ker iskanje obrazov ni tako časovno pomembno, ga mora model prepoznati vsaj 10x, kjer mora med vsako prepoznavo miniti manj kot 0.2 sekundi, da pošlje prepoznan obraz naprej.

1.2 Prepoznavanje cilindorv