

Predefinisani projekat za ocenu 9/10

Arsenije Degenek RA59-2014

Fakultet tehničkih nauka

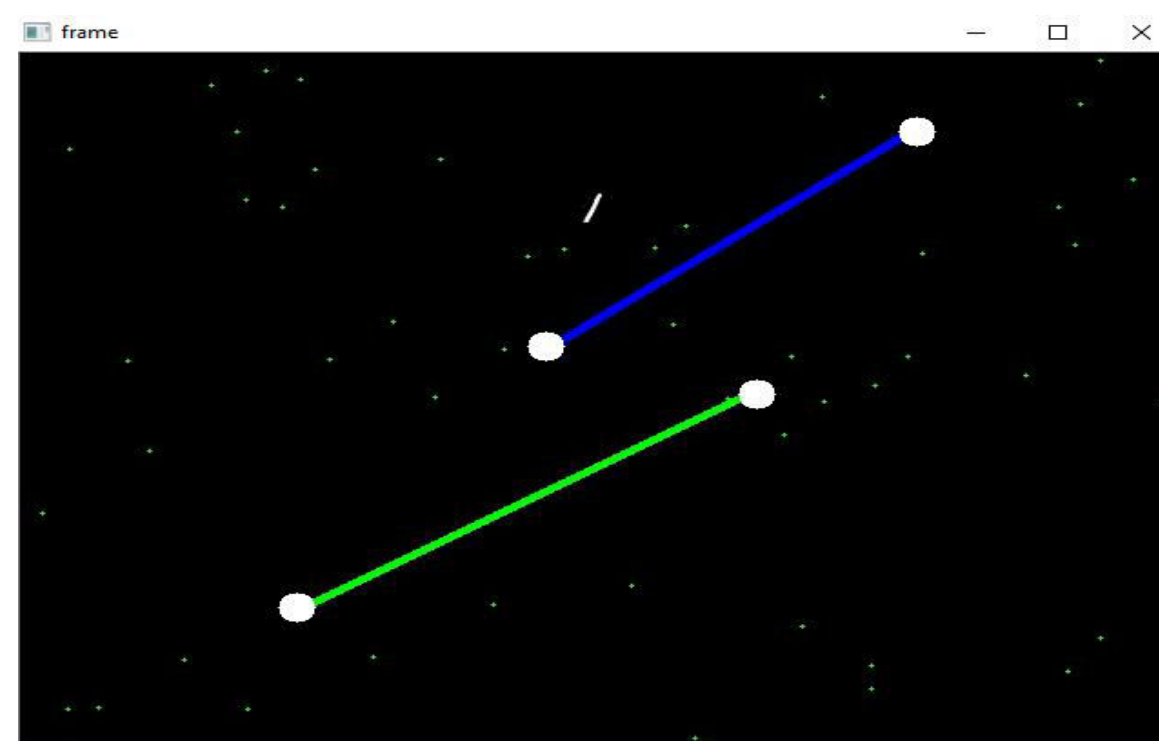
Opis problema

Video snimak sadrži plavu i zelenu liniju koje se ne pomeraju. Cifre predstavljaju pokretne komponente snimka. Potrebno je prepoznati cifre koje prođu ispod linija. One koje prođu ispod plave treba sabrati, a one koje prođu ispod zelene treba oduzeti od ukupnog zbira. Cifre mogu da se preklapaju na bilo kojoj tački tokom svoje putanje. Potrebno je postići tačnost od 95%.

Metod detektovanja linija

Za detektovanje linija na snimku korišćen je metod Hough-ove transformacije. Postupak je isti za obe linije, i sastoji se iz sledećih koraka:

- Uzima se jedan frejm sa snimka (nebitno je koji, pošto se linije ne kreću)
- Metodom otvaranja (prvo erozija, pa dilacija), uklanja se šum sa snimka u vidu belih tačkica
- Nuliraju se RB komponente slike (za zelenu liniju), odnosno RG komponente (za plavu liniju) te na frejmu ostaje samo tražena linija i cifre obojene njenom bojom
- Frejm se konvertuje u nijanse sivih (grayscale) radi lakše obrade
- Vršiti se detekcija linija upotrebom Canny Edge Detection algoritma, koji vraća frejm sa iscrtanim ivicama
- Tako dobijeni frejm prosleđuje se metodi HoughLinesP, koja će vrtatiti koordinate početne i krajnje tačke svake linije koja je zadovoljila prosleđene parametre
- Pošto će HoughLinesP detektovati posmatrnu liniju iz više manjih linija, potrebno je naći minimalne početne i maksimalne krajnje koordinate, pri čemu se konačno dobijaju koordinate početka i kraja tražene linije (slika 1.1)

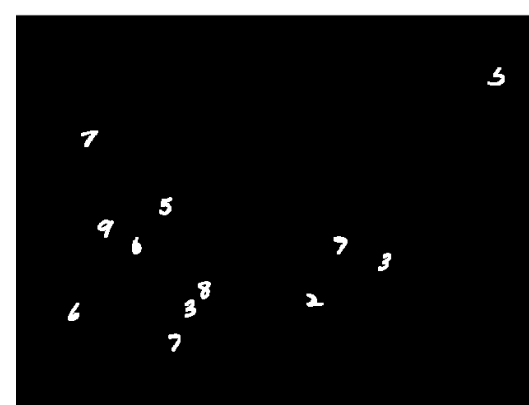


1.1 Detektovane ivice linija na snimku

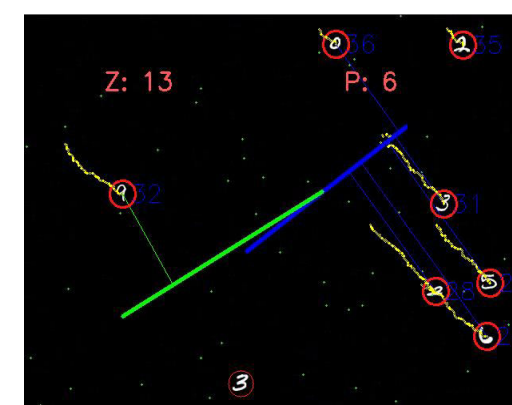
Izdvajanje cifara sa snimka

Da bismo zapamtili sve cifre koje prolaze ispod linija, potrebno je analizirati svaki frejm video snimka. Postupak pronalaženja cifara se sastoji iz sledećeg:

- Konverzija svakog frejma u nijanse sivih (grayscale), nakon čega se vrši binarizacija istog pozivom metode threshold sa definisanim pragom, tako da na frejmu ostaju samo cifre (slika 2.1)
- Svaka nova cifra koja se pojavi na frejmu dobija svoj id i prati se njeno kretanje dok god ne nestane sa snimka (slika 2.2)

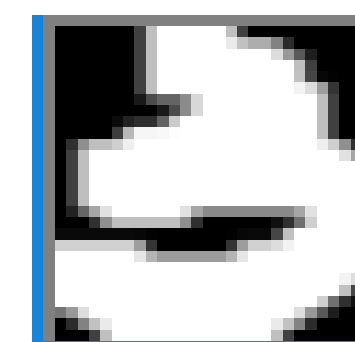


2.1 Binarizovana slika

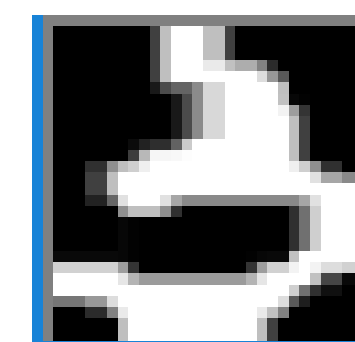


2.2 Praćenje cifara

- Potrebno je iseći (crop) cifru sa frejma na kom se prvi put pojavila i odraditi resize na dimenzije 28x28, da bi mogla biti prepoznata od strane knn-a
- Takođe, potrebno je odraditi i eroziju nad isečenom slikom radi boljeg pogađanja (slike 2.3, 2.4)
- Za svaki frejm se proverava da li je udaljenost centar cifre od linija manja od 9 piksela (pozivom funkcije pnt2line), te ukoliko jeste, detektovali smo cifru koja je prošla ispod linije



2.3 Cifra pre erozije



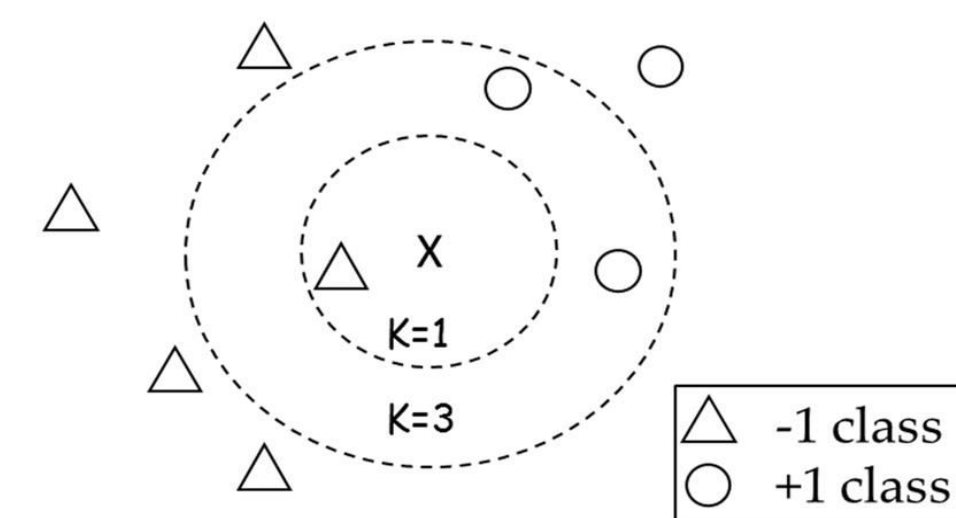
2.4 Cifra posle erozije

Rad sa KNN algoritmom

Da bi se postiglo što bolje prepoznavanje cifara, korišćen je K-Nearest Neighbour algoritam za klasifikaciju. (Slika 3.1) Sam algoritam zavisi od dve stvari:

- Metrika koja se koristi za izračunavanje udaljenosti između 2 tačke
- Vrednost K broja suseda koje treba razmotriti

Za obuku je korišćen MNIST dataset koji sadrži 70 000 ručno pisanih cifara. S obzirom na to da su cifre iz dataseta centrirane i udaljene od ivica slike, bilo ih je potrebno iseći (crop) i odraditi resize na dimenzije 28x28 da bi izgledale isto kao i isečene cifre iz video snimka.



3.1 KNN

Zaključak

Python programski jezik, zajedno sa bibliotekama koje nudi pokazao se kao izuzetno dobar alat za rešavanje ovog tipa problema, pre svega zahvaljujući olakšanoj obradi slika, ali i rada sa KNN-om. Cifre koje se preklapaju su uticale na procenat tačnosti rešenja, jer zaklanjaju jedna drugu pa ih je teško prepoznati. Ako se to zanemari, postignuti rezultati su skroz zadovoljavajući.