

Detekcija i klasifikacija brojeva iz MNIST dataset-a na video zapisima

Dušan Svilarković

asistent: Miroslav Kondić

Uvod

Zadatak ovog projekta je praćenje kretanja brojeva iz poznatog MNIST dataset-a po video zapisu, pri čemu kad se detektuje prolazak broja kroz neku od dve linije je potrebno odgovarajuće sabrati ili oduzeti tu vrednost od ukupne vrednosti čitavog video zapisa. Detektovanje linija potrebno je uraditi Hough transformacijom.

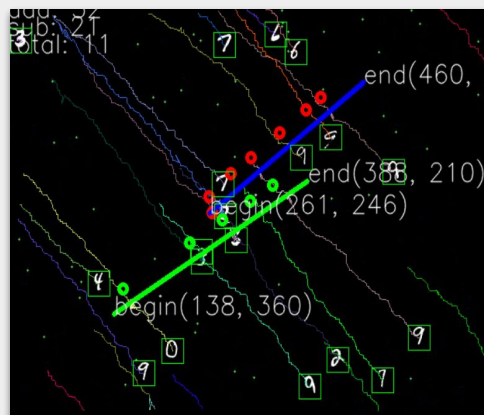
Okruženje korišćeno

Jupyter Notebook i Python sa OpenCV, Scipy, Keras (TF backend) bibliotekama.

Postupak

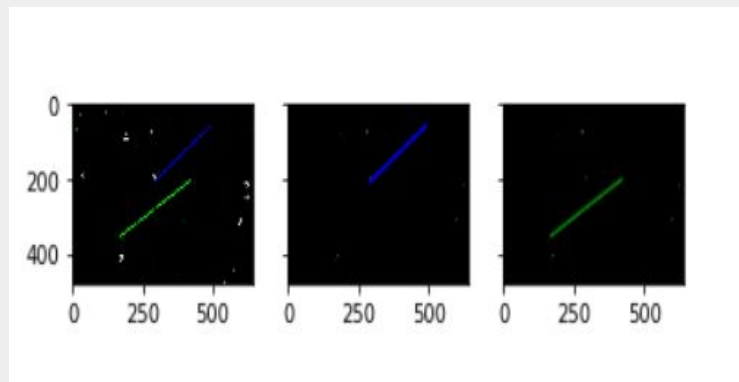
Sam dolazak do detekcije brojeva sastojao se iz:

1. Izdvajanja **linija** korišćenjem Hough transformacije i izdvajanjem kanala boje po HSV..
2. Dilacijom, Otsu i maskiranjem boje dolazimo do osamostaljenja **brojeva**.
3. Korišćenjem **findContours** iz OpenCV dolazimo do kontura koje predstavljaju **brojeve** i mapiramo ih u 28x28 prostor.
4. Ako je zadato kriterijumom, klasifikujemo brojeve primenom **konvolucione neuronske mreže** (CNN).
5. Ponavljamo korake od 1-4 za svaki frejm i čuvamo istorijat za svaku putanju da bi adekvatno detektovali prolaz kroz **liniju**.



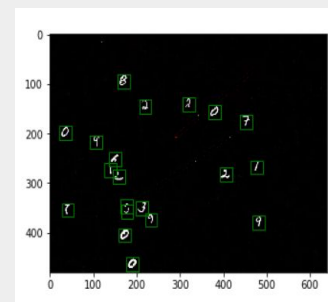
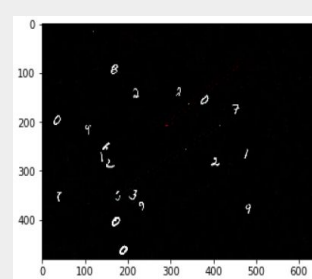
Izdvajanje linija

Izdvajanje linija vrši se korišćenjem **Hough transformacije**. Ovde nema nekog posebnog dodatnog objašnjenja, sem da se kao „izazov” može posmatrati izdvajanje po kanalu, jer se u ovom slučaju nalaze dve različite linije, te za svaku ponaosob treba po jednom color kanalu tražiti transformaciju, plavom za „sabiračku” liniju i zelenom za „oduzimačku” liniju.



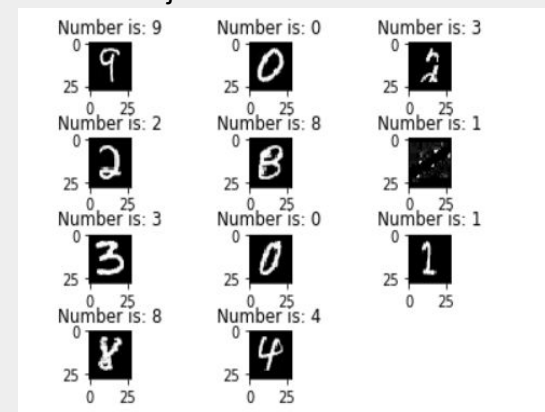
Izdvajanje brojeva

Izdvajanje se jednostavno obavlja, s obzirom da su brojevi predstavljeni belom bojom, a u boji (zelenoj) se nalaze šumovi i „oduzimačka” linija, a u plavoj „sabiračka” linija. Korišćenjem **plave** i **zelenе** maske uklanjaju se boje, a korišćenjem **findContours** iz OpenCV biblioteke se izdvajaju brojevi.



Klasifikacija brojeva

Nakon što su adekvatno locirani brojevi na statičkoj slici, u zavisnosti od kriterijuma, tj. da li je pređena linija, potrebno je izračunati klasifikatorom, u ovom slučaju konvolucionom neuronskom mrežom, koji je broj u pitanju. Broj se klasifikuje tako što se zaokruži prostor 28x28 oko nje na zamrznutom frejmu, ponovo izvrši findContours da bi se centrirao broj i uklonio okolni šum sa slike, i tek onda izvrši konačna klasifikacija.



Logika vremena sabiranja i oduzimanja brojeva

S obzirom da se vremenom menjaju pozicije brojeva, čuvaju se **putanje/istorijati** kretanja, pa se za svaki nađeni broj na svakom frejmu određuje njegov **istorijat** tako što se gleda najbolji kandidat na osnovu daljine putanje i poslednje pojave putanje. Kada broj pređe liniju, uzeće se u istorijatu njegove putanje frejm kada se prvi put broj pojavio i iz njega izvući centroid i kako je broj tada izgledao i tako konačno klasifikovati.

Utisci i zaključci

Na kraju, izazov u ovom problemu predstavlja problem preklapanja brojeva i linearnog kretanja sa šumom, koji pored toga što a) onemogućava precizan način određivanja putanje kretanja, onemogućava i b) adekvatnu odluku za izbor vremena u kojem je najbolje broj iskoristiti i klasifikovati.

Predlog rešenja odluke kretanja se može koliko toliko odrediti **linearnom regresijom** putanje, ili primenom **Kalmanovog filtera**, međutim sam problem preklapanja nije moguće rešiti ako je preklapanje brojeva izuzetno izraženo tokom čitave istorije kretanja broja.