

# Prepoznavanje i prebrojavanje pokretnih cifara

Problem koji se resava:

Video zapis poseduje dve pokretnе linije koje je potrebno detektovati Hough transformacijom.

Rucno pisane cifre prolaze iza pokretnih linija.

Cifre koje prodju ispod plave linije treba sabrati, a cifre koje prediju ispod zelene linije treba oduzeti od konačnog rezultata.

U video zapisu postoje sluzajevi preklapajućih cifara koje istovremeno prelaze preko linije, a nalaze se jedna preko druge u određenoj meri.

Detectacija linija

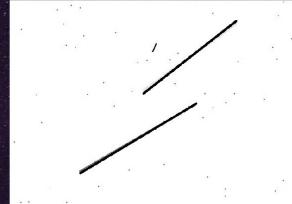
Posto su linije nepokretne na video snimku, dovoljno je uzeti prvi frame i njega obraditi.

Iz slike prvog frejma, generisu se 2 nove slike koje predstavljaju sliku samo sa zelenom linijom, odnosno sliku samo sa plavom linjom.

Nakon toga sledi dalja obrada slike zarad Hough transformacije.

Kada je slika spremna, uz pomoc Hough transformacije mi dobijamo par tacaka koje predstavljaju nasu traženu liniju.

1. Poziva se probabilistička Hough transformacija
2. Rezultat Hough transformacije ce nam dati niz linija, jer svice nase linje nisu savrseno ravne (već su razbijene)
3. Biramo tacke kod kojih jedna predstavlja tacku najmanju po Y-osi, a druga najvecu po Y-osi



Isrtane linije na osnovu para tacaka koje su dobijene Hough transformacijom

1. Konvertovanje u RGB iz BGR modela boja
2. Pretvarjanje u gray Scale
3. Pretvarjanje u binarni oblik (preko thresholda)
4. Upotreba erozije, radi poboljšanja tачnosti

1. Matricu 28x28 pretvaramo u niz od 784 elemenata.
2. Svi dan tog niza skaliramo na vrednost [0,1].

- Sama transformacija se izvrsava iz nekoliko koraka:
1. Slika se pretvara u HSV model boja
  2. Prikupljaju se OpenCV-a predviđene opće RGB piksele koji je narušili raspored
  3. Na kraju slika se u pomoći dilacije, odnosno erozije delje doteže kako bi naredni koraci bili uspešniji

1. Slika se pretvara u binarni oblik uz pomoć Threshold funkcije.
2. Nad dobijenim oblikom koristi se 'Canny Edge Detector' za detekciju ivica.
3. Gausovo zamacenje (radi omekšavanja ivica.)

Prepoznavanje brojeva

Za prepoznavanje brojeva sa slikom koriscena je **neuronska mreža**.

Sam postupak izdvajanja sličica brojeva sa jednog frame-a pocinje prvo sa **obradom slike frejma**.

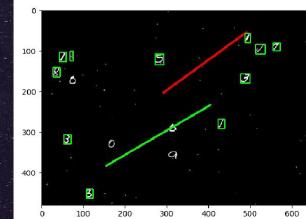
Na osnovu obradjene slike **pronalaze se sve konture brojeva**.

Posto sve dobijene konture nisu dimenzija 28x28, zarad neuronske mreze mi to moramo da napravimo (kako bi tacnost bila veca), te cemo svaku sliku prosiriti sa neophodnim brojem crnih piksela sa svih strana, tako da originalna proporcija slike ostane netaknuta, a da broj takodje ostane i centriran na novoj slici.

Tu nije kraj. Sliku moramo da **pretvorimo** u oblik koji mozemo da prosledimo kao ulaz za neuronsku mrežu.

Finalno, prosledjujemo ulaz u neuronsku mrežu i kao povratnu vrednost dobijamo broj (0-9) koji nam predstavlja vrednost sa slike.

Povratnu vrednost i kordinate konture prosledjujemo u novi objekat Broj.



Detectovanje brojeva sa trenutnog frejma.  
**(Napomena:** Ovaj algoritam za detectovanje brojeva ima za posledicu da se nule ne detektuju, ali to ne pravi problem, posto one ne menjaju rezultat.)

Neuronska mreža koja za ulaz ima 784 neurona (svi pikseli slike veličine 28x28), tri skrivena sloja od po 256 neurona i jedan izlazni sloj sa 10 neurona (cifre 0-9).

Kao aktivacione funkcije koristile su se 'relu' i 'softmax', jer su pokazale najbolju praksu.

Za trening skup se koristio obradjeni MNIST dataset rucno pisanih brojeva. Radi bolje tacnosti prepoznavanja brojeva sa video snimka, na ulaz za treniranje neuronske mreže za svaki broj iz dataseta ubacena je i njegova kopija koja je modifikovana dilacijom, odnosno erozijom.

Sve konture se nalaze uz pomoć funkcije Open CV-a.

Nakon sto dobijemo povratnu vrednost radimo filter nad pronadjenim konturama.

Biramo samo one koje imaju visinu između 15 px i 30 px i širinu manju od 30 px (jer samo takve konture mogu predstavljati neokrenute slike broja) i takođe kontura ne smete biti prekopljeni nekom od linija, ili drugom konturonu nekog broja.

Detectacija prelaza:

Za detekciju prelaza se koristi prava kretanja broja.

Svaki broj u sebi ima 4 atributa: vrednost, kordinate sredisne tacke, kordinate prvog pojавljivanja, kordinate drugog pojавljivanja.

Svakom broju koji se "prije put" pojavljuje na frejmu dodeljuju se kordinate prvog pojавljivanja, tako sto se preuzmu kordinate sredisne tacke konture broja.

Ukoliko broj ima vec unete kordinate prvog pojавljivanja, a nema upisane kordinate drugog pojавljivanja, njemu se taj atribut puni sa vrednostima kordinata sredisne tacke za trenutni frej.

Cim smo uneli vrednosti u oba atributa pojavitivanja i potvrdili da taj broj nije vec registrovan, vrsi se ispitivanje da li prava koja prolazi kroz 2 tacke sece duz za sabiranje ili duz za oduzimanje. Shodno tome, rezultat se azurira.

(Napomena: Ovo ne vazi za specijalan slučaj kod brojeva koji su detektovani u prvom frejmu, a nalaze se ispod obe linije.)

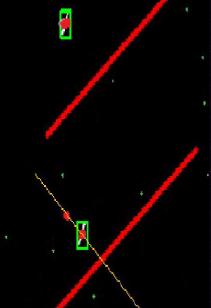
Za cuvanje informacija o kordinatama pojavitivanja koristi se mehanizam **pracenja brojeva**.

Pracenje brojeva je omoguceno uz pomoc 2 niza : liste prethodnih brojeva i liste trenutnih brojeva.

U prvi iteraciji te 2 liste ce biti jednake, a na sledećoj se iteraciji ce se iterirati kroz listu trenutnih brojeva i traziti da li se ona nalazi u listi prethodnih brojeva. Ako se nalazi, svii vazni atributi se prebacuju iz starog broja u novi.

Provera jednakosti se vrsti iz 2 uslova:  
1. Da je vrednost 2 broja jednaka  
2. Da je euklidsko rastojanje izmedju 2 opsegova 10-60, (manje ne sme biti jer se pomeraju za 30 frejmova)

Prvo pojavljivanje



Druge pojavljivanje i provlačenje orave kretanja

Ideja detekcije prelaska preko linije sabiranja za dati broj.