

Prepoznavanje i prebrojavanje pokretnih cifara

Problem koji se resava:

Video zapis poseduje dve pokretne 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 predju ispod zelene linije treba oduzeti od konacnog rezultata.

U video zapisu postoje slucajevi preklapajucih cifara koje istovremeno prelaze preko linije, a nalaze se jedna preko druge u odredjenoj meri.

Detekcija 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 linijom.

Nakon toga sledi dalja obrada slike zarad Hough transformacije.

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

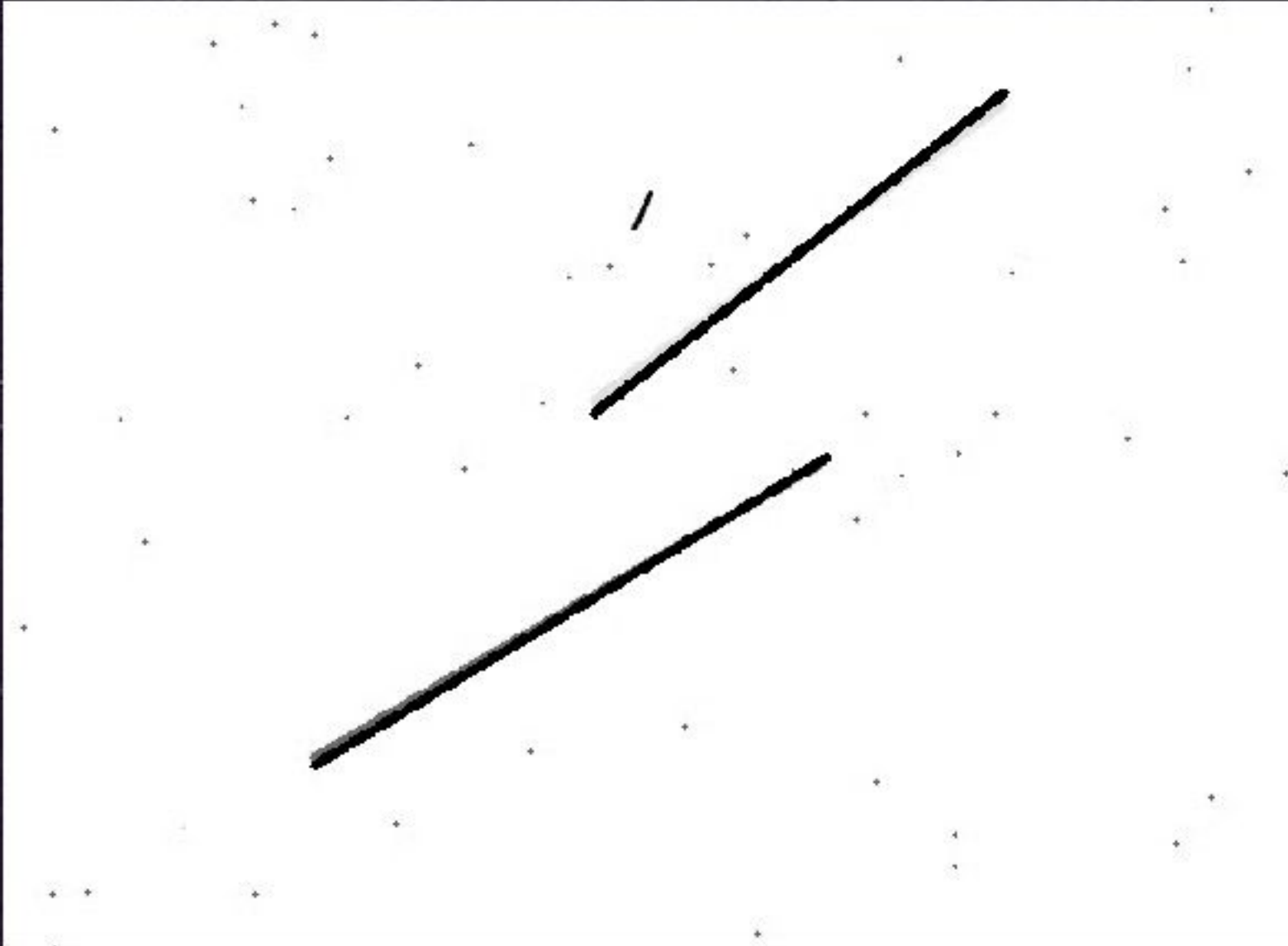
1. Poziva se probalisticka Hough transformacija
2. Rezultat Hough transformacije ce nam dati niz linija, jer ivice nasih linija nisu savrseno ravne (vec su nazubljene)
3. Biramo tacke kod kojih jedna predstavlja tacku najmanju po Y-osi, a druga najveću po Y-osi

1. Konvertovanje u RGB iz BGR modela boja
2. Pretvaranje u gray Scale
3. Pretvaranje u binarni oblik (preko tresholda)
4. Upotreba erozije, radi poboljsanja tacnosti

1. Matricu 28x28 pretvaramo u niz od 784 elementa.
2. Svaki clan tog niza skaliramo na vrednost [0,1].

- Sama transformacija se izvrsava iz nekoliko koraka:
1. Slika se pretvara u HSV model boja
 2. Preko metode Open CV-a izvlacimo opseg RGB piksela koji je nama potreban
 3. Na kraju slika se uz pomoc dilacije, odnosno erozije dalje doteruje kako bi naredni koraci bili uspesniji

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



Iscrtane linije na osnovu para tacaka koje su dobijene Hough transformacijom

Prepoznavanje brojeva

Za prepoznavanje brojeva sa slika korisca je **neuronska mreza**.

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

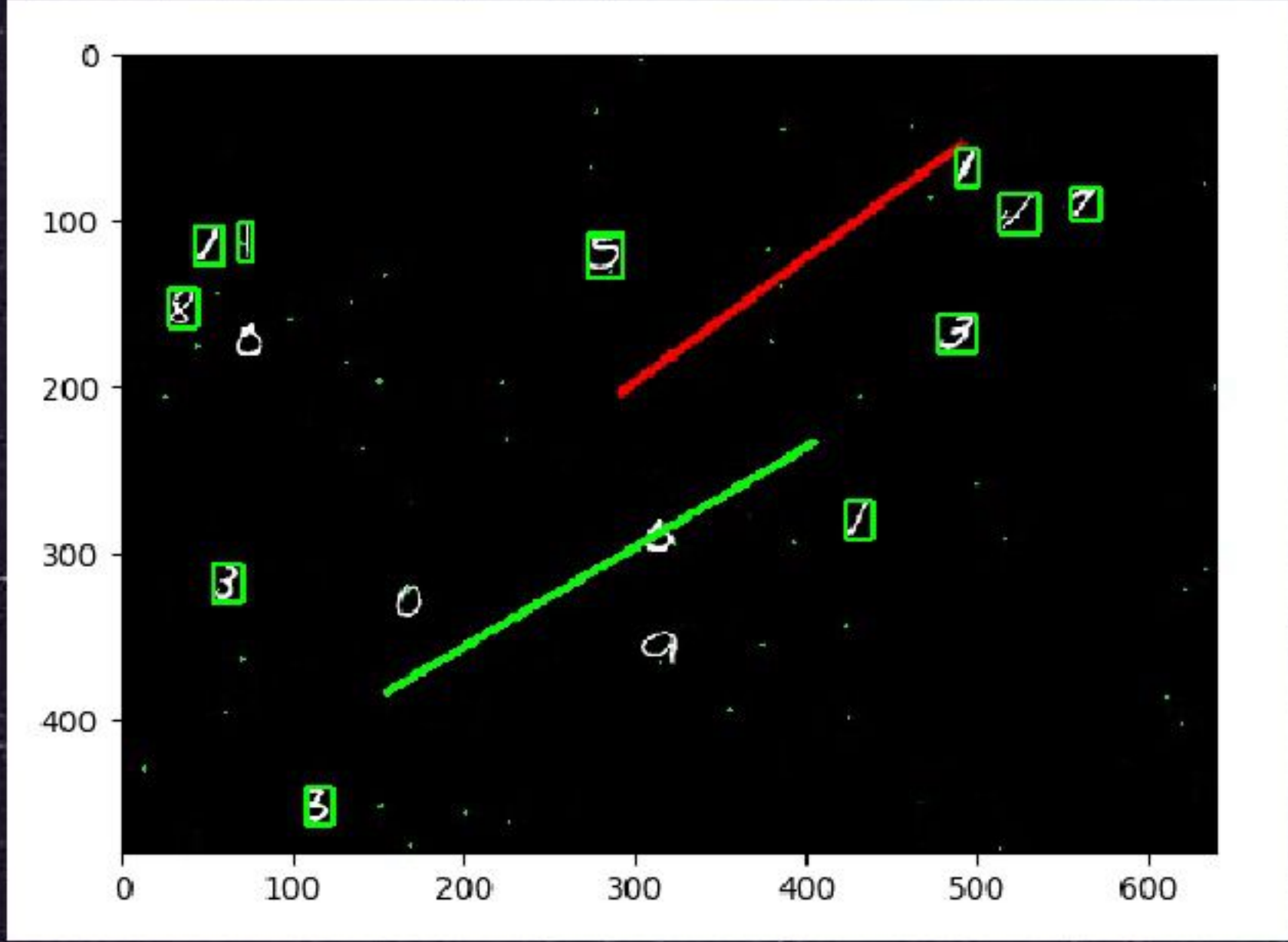
Na osnovu obradjene slike **pronaleze se sve konture brojeva**.

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 mrezu.

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

Povratnu vrednost i kordinate konture prosledjujemo u novi objekat Broj.



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

Neuronska mreza koja za ulaz ima 784 neurona (svi pikseli slike velicine 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 mreze za svaki broj iz dataseta ubacena je i njegova kopija koja je modifikovana dilacijom, odnosno erozijom.

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

Nakon sto dobijemo povratnu vrednost radimo filter nad pronadjenim konturama.

Biramo samo one koje imaju visinu izmedju 15 px i 30 px i sirinu manju od 30 px (jer samo takve konture mogu predstavljati neokrzljenu sliku broja) i takodje kontura ne sme biti preklapljena nekom od linija, ili drugom konturom nekog broja.

Detekcija prelaza:

Za detekciju prelaza se koristi prava kretanja broja.

Svaki broj u sebi ima 4 atributa: vrednost, kordinate sredisnje tacke, kordinate prvog pojavljivanja, kordinate drugog pojavljivanja.

Kordinate prvog pojavljivanja se pune kordinatama sredisnje tacke broja prilikom prvog pojavljivanja broja u oblasti do 100 px po X ili Y osi. (Time smo obezbedili da ukoliko se broj detektuje nakon sto je predhodno bio prekriven linijom ili nekim brojem, a nemamo informaciju o njemu u listi brojeva predhodnog frejma, da ga ne racuna vise puta.)

(Napomena: Izuzetak ovome je inicijalno detektovanje brojeva sa prvog frejma, gde se SVAKOM broju puni vrednost kordinata prvog pojavljivanja.)

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

Cim smo uneli vrednosti u oba atributa pojavljivanja, vrsi se ispitivanje da li prava koja prolazi kroz te 2 tacke sece duz za sabiranje ili duz za oduzimanje. Shodno tome, rezultat se azurira.

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

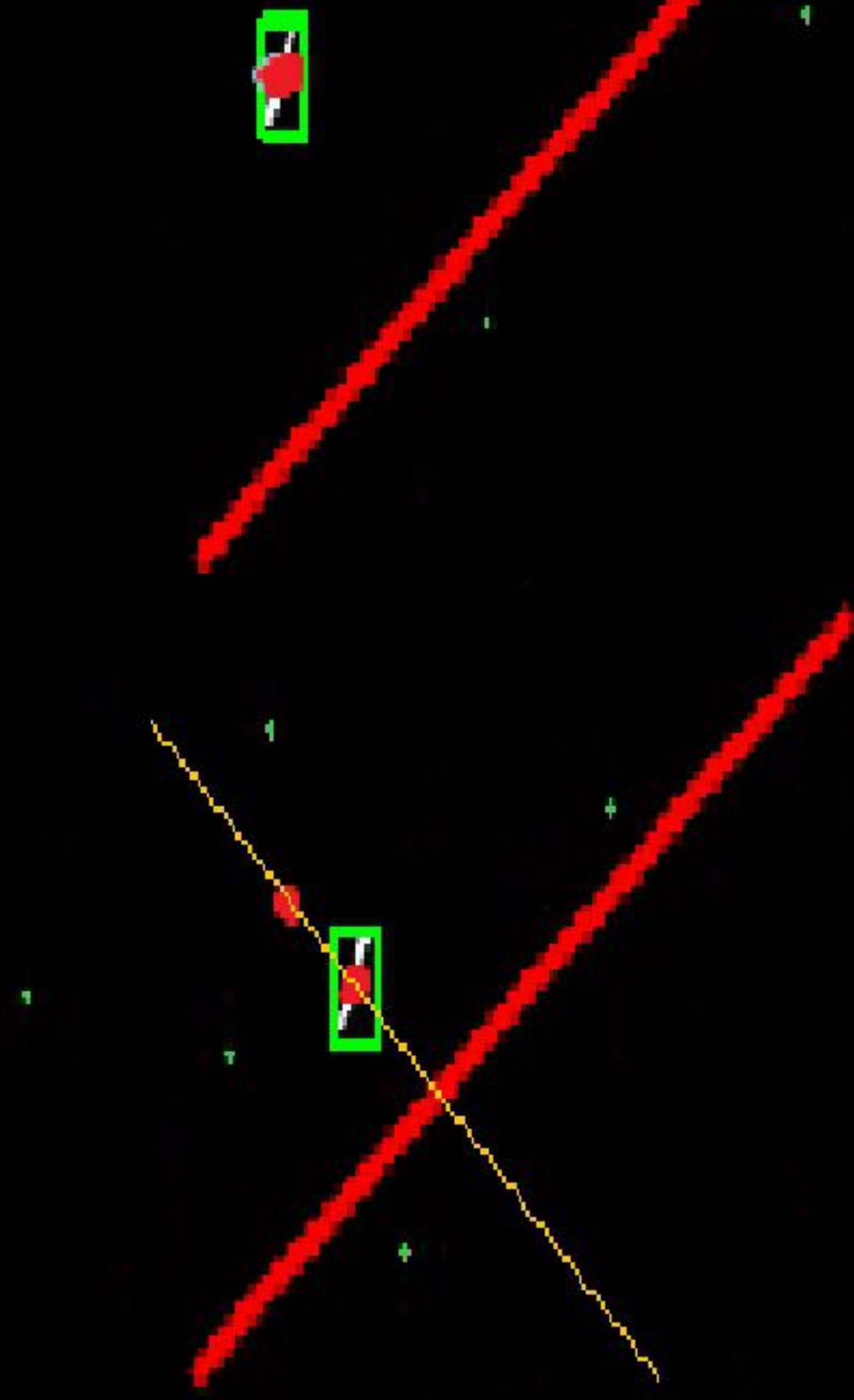
Za cuvanje informacija o kordiantama pojavljivanja koristi se mehanizam **pracenja brojeva**.

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

U prvoj iteraciji te 2 liste ce biti jednake, a na svakoj sledecoj ce se iterirati kroz listu trenutnih brojeva i traziti da li se ona nalazi u listi predhodnih brojeva. Ako se nalazi, svi vazni atributi se prebacuju iz starog broja u novi.

Provera jednakosti se vrsi iz 2 uslova:
1. Da je vrednost 2 broja jednaka
2. Da je euklidsko rastojanje izmedju njih u opsegu 10-60. (manje ne sme biti jer se pomeramo za 30 frejmova)

prvo pojavljivanje



Drugo pojavljivanje i provlacenie prave kretania

Ideja detekcije prelaska preko linije sabiranja za dati broj.