

Detekcija i prepoznavanje cifara sa slike

1. Detekcija linija korišćenjem Hough transformacije

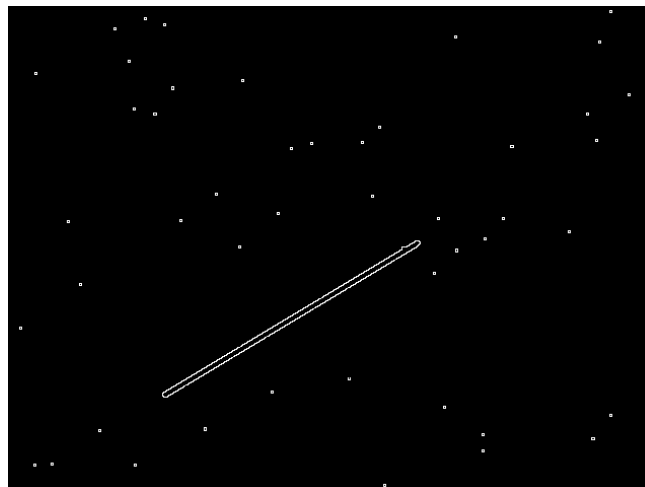
S obzirom da su linije statične dovoljno je samo u jednom frejmu izračunati njihove krajnje tačke. U datom frejmu najpre vršimo njegovu obradu radi dobijanja što preciznijih koordinata krajeva tačaka kada frejm prosledimo metodi **cv2.HoughLinesP()**.

Obradu frejma vršimo tako što prvo konvertujemo model boja u HSV a zatim definišemo donju i gornju granicu opsega nijansi zelene i plave boje predstavljene u ovom modelu boja. Te opsege i sam frejm sa HSV modelom prosleđujemo metodi **cv2.inRange()** koja će nam vratiti binarnu masku veličine frejma na kojoj se nalazi plava, odnosno zelena linija.

Kada imamo maske za obe linije primenjujemo ih na frejm konvertovan u grayscale model koji je potreban za dalju obradu. Na slici je prikazan primer sa zelenom linijom na kojem se vidi i šum, međutim on će biti zanemaren prilikom Hough transformacije.



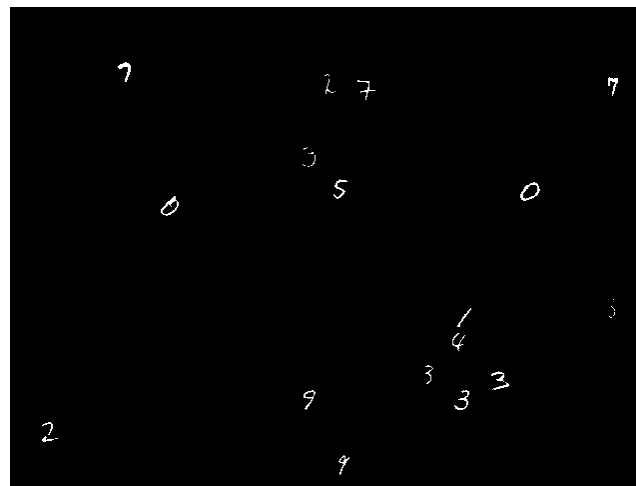
Sledeći korak pre transformacije je detekcija ivica na frejmu upotrebom **cv2.Canny()** metode. Ova metoda će na osnovu prosleđenog frejma u grayscale modelu vratiti binarnu sliku sa detektovanim ivicama



Frejm u ovom obliku sada možemo proslediti metodi **cv2.HoughLinesP()**. Ova metoda će na osnovu prosleđenih parametara vratiti niz početnih i krajnjih koordinata objekata koji su detektovani kao linije. Sa slike se može primetiti da ova metoda neće detektovati liniju kao celinu zbog postojanja gornje i donje ivice linije kao i zbog rezolucije. Da bismo dobili stvarne koordinate krajeva linije u nizu koordinata pronalazimo krajnju donju-levu i krajnju gornju-desnu koordinatu. Ove dve koordinate predstavljaju koordinate krajeva tačaka

1. Obrada frejma i izdvajanje regiona od interesa

Nad svakim frejmom potrebno je izvršiti određene transformacije kako bi se pripremio izdvajanje regiona od interesa, tj. cifara . Prvo je potrebno ukloniti šum i linije. S obzirom da su cifre bele a šum i linije zelene i plave boje, možemo primeniti filter boja gde će donji prag biti svetlija nijansa sive a gornji prag bela. Filter primenjujemo upotrebom **cv2.inRange()** metode koja kao parametre prima ova dva praga zadata kao rgb vrednosti.



Nakon primene filtera na mestima gde su cifre bile prekrivene šumom doslo je do prekida. Da bismo popunili te praznine primenjujemo dilataciju korišćenjem **cv2.dilate()** metode. Ovim smo obezbedili da u daljoj obradi frejma cifra ne bude izdvojena u dva ili više regiona.

Kako bismo dobili niz isečaka na kojima se nalaze cifre prvo pozivamo metodu **ndimage.label()** koja sa prosleđene binarne slike kao rezultat vraća sliku gde svaki objekat ima jedinstvenu labelu. U sledećem koraku nad označenom slikom pozivamo **ndimage.find_objects()** od koje dobijamo željeni niz svih isečaka sa ciframa.

2. Obrada niza isečaka i prepoznavanje cifara

Za svaki isečak koji smo dobili **find_objects** metodom računamo centar regiona kao i veličinu. Ukoliko je ipak došlo do detekcije više objekata za jednu cifru uklanjamo zanemarujemo objekte manje od očekivanog.

Da bismo tokom obrade frejmova imali informacije o objektima i njihovom kretanju, za svaki od objekata formiramo rečnik i beležimo informacije o jedinstvenom identifikacionom broju, koordinatama centra, veličini, brojača trenutnog frejma, niza koordinata centara koje je objekat imao u prethodnim frejmovima(istorija).

Postoje dve načina na koji tretiramo svaki objekat kada je u pitanju niz rečnika detektovanih objekata. Za svaki objekat trenutnog frejma potrebno je zaključiti da li se on prvi put pojavio na frejmu ili je u ranijim frejmovima već detektovan. Da bismo to zaključili koristimo metodu **inRange()** koja za prosleđeni objekat i vrednost koja predstavlja očekivani pomeraj cifre na svakom frejmu uvećan za određenu toleranciju. Metoda kao rezultat vraća listu elemenata koji su se nalazili u okolini posledenog objekta tražeći ih u rečniku rečnika(elementa) koji su zabeleženi do trenutnog frejma. Ukoliko je ta lista prazna zaključujemo da je se objekat prvi put pojavljuje u trenutnom frejmu, vršimo prepoznavanje same cifre i dodajemo kreirani rečnik analiziranog objekta u niz elemenata. Ukoliko je dužina liste koju je metoda vratila jednaka jedinici pretpostavljamo da je jedini element te liste baš trenutno posmatrani objekat, pa ažuriramo taj element sa informacijama o trenutnom centru objekta i dodajemo taj centar u istorijske podatke elementa.

3. Detektovanje prelaska objekta preko linije

Nakon ažuriranja niza detektovanih elemenata, za svaki element liste potrebno je izračunati njegovu udaljenost od linije i ažurirati vrednost rezultujuće sume ukoliko je element na liniji.

Za izračunavanje udaljenosti i koristimo funkciju **pnt2line()** kojoj prosleđujemo centar elementa i koordinate početka i završetka linije. Ovo izračunavanje vršimo za obe linije na frejmu i ako je centar elementa na dovoljno maloj udaljenosti od linije dodajemo, odnosno oduzimamo od sume prepoznatu vrednost elementa, a za element beležimo da je prešao liniju kako ga ne bi uzimali u obzir u narednim frejmovima.