Predefinisani projekat za ocene 9 i 10

Filip Petrović RA49-2014

Soft kompjuting 2017-2018 [E2]

Tema projekta

Cilj ovog projekta jeste obrada video zapisa. Za dati video potrebno je detektovati dve linije koje se ne pomeraju , od kojih je jedna plava i jedna zelena. U toku videa ručno napisane cifre se pojavljuju i prolaze kroz linije. Potrebno je uočiti koliziju objekta (broja) i linija. Kada se dogodi kolizija potrebno je na osnovu slike broja, koristeći KNN algoritam, prepoznati o kom broju se radi i taj broj sabrati/oduzeti od ukupne sume kad pređe preko plave/zelene linije.

Problemi i korišćene metode

U ovom projektu razlikujemo 4 osnovna problema:

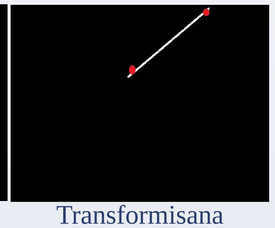
1. Detekcija linija korišćenjem Hough transformacije
2. Detekcija regiona brojeva sa tekućeg frejma korišćenjem SciPy python biblioteke (ndimage)
3. Detekcija kolizije objekta (broja) sa linijom računajući distancu od centra objekta do najbliže tačke na bližoj od dve linije ili korišćenjem linearne regresije.
4. Prepoznavanje vrednosti broja sa slike koristeći razne metode obrade slike iz OpenCV biblioteke, KNN-algoritma, i Mnist dataseta.

Detekcija linija

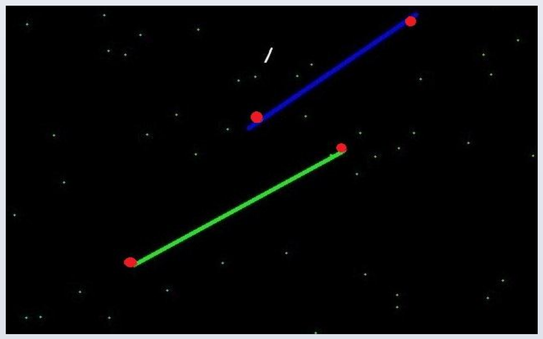
Uzeta je slika prvog frejma videa. Ta slika se filtrira tako da se dobiju dve crno bele slike. Jedna prikazuje poziciju zelene linije u frejmu a druga prikazuje poziciju plave linije u frejmu. Obe slike se prosleđuju Canny Edge algoritmu koji nalazi ivice, pa se nakon toga slike prosleđuju HoughLinesP funkciji koja vraća temena linija sa slike.

Uzimajući u obzir da se na svakoj od slika nalazi samo jedna jedina linija, potrebno je samo uporediti svaku od temena linija i izvući minimalnu i maksimalnu vrednost po X osi da bi se dobila temena linije.

Transformisana slika zelene linije sa obeleženim ivicama



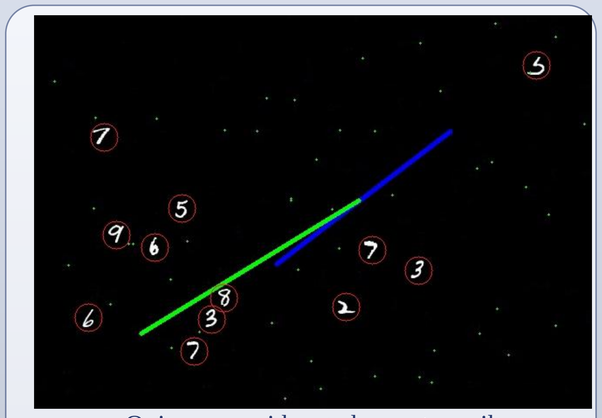
Transformisana slika zelene linije sa obeleženim ivicama



Izgled trenutnog frejma sa obeleženim ivicama obe linije

Detekcija regiona brojeva

Za detekciju regiona brojeva na tekućem frejmu koristimo ndimage iz SciPy biblioteke koji sadrži veliki broj funkcija za obradu slike. Naime, za prosleđeni trenutni frejm koristimo metodu label da bi se svaki prepoznati objekt sa slike označio jedinstvenom int vrednošću. Pa se nakon toga poziva metoda find_objects nad labeliranim frejmom. Metoda find_objects vraća listu objekata sa informacijama o njihovim pozicijama i centru. Imajući poziciju svakog objekta (broja) na frejmu kao i njegove dimenzije, možemo pratiti poziciju tog broja i detektovati kolizju ako do nje dođe.



Opisan crveni krug oko centra svih detektovanih brojeva

Detekcija kolizije broja i linije

Prvi način

Znajući gde se nalazi centar broja kao i ivice obe linije koristimo metodu point2line koja vraća distancu od centra broja do najbliže tačke na liniji. Kako se brojevi kreću za svaki se računa distanca od najbliže tačke na obe linije. Ako dobijena distanca manja od potrebne, to ćemo detektovati kao koliziju i sabrati/oduzeti dati broj.

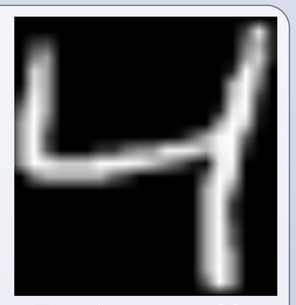
Drugi način

Za svaki od brojeva se vodi evidencija o tome koje pozicije je on tokom videa imao. Na osnovu tih tačaka možemo naći regresionu pravu koja predstavlja aproksimaciju putanje datog broja. U slučaju da se regresiona prava seče sa nekom od linija , to se detektuje kao kolizija i onda sledi sabiranje/oduzimanje. Na ovaj način se prepoznaju i oni brojevi koje neki drugi broj preklopi tačno na pravoj, tako imamo veću tačnost krajnje sume.

Prepoznavanje vrednosti broja sa slike

Kada se preuzmu podaci iz MNIST-a, nad tim podacima se moraju izvršiti transformacije da bi one bile upotrebljive. Cilj je da dobijemo sliku formata 28x28 koja nema suvišnog prostora već je cela „prekrivena“ brojem.

Slika iz MNIST-a



Transformisana slika

Nad skupom transformisanih slika iz MNIST-a se obučava KNN sa brojem komsija k = 1 jer za taj broj komšija dobijamo najbolje rezultate. Svaki broj koji uočimo na videu ima svoj centar i svoje dimenzije, sledi da ga možemo iseći sa slike. Da bi KNN mogao da prepozna o kojem broju je reč, moramo i isečenu sliku dodatno da transformišemo da bi odgovarala formatu slike koji se koristi u MNIST-u. Kada se izvrše potrebne transformacije, isečena i transformisana slika se prosleđuje KNN-u koji vraća vrednost broja sa slike.

Rezultati

Rezultat se ogleda u tačnosti izračunate sume za svaki video. Uzmajući u obzir da Hough-ova transformacija ne daje idealne rezultate i da se neretko dešava da je jedan broj preklopljen drugim od prvog njegovog pojavljvanja na videu, rezultati nikda ne mogu biti tačni 100% ako je rešenje implementirano prethodno opisani način. Takođe za obuku KNN algoritma se svaki put uzima random deo iz data seta pa rezultati nikada nisu isti.

Literatura

NumPy – korisnička dokumentacija

SciPy – korisnička dokumentcija

OpenCV – korisnička dokumentacija

https://github.com/squeakus/octree/blob/master/vectors.py

Materijali sa vežbi sa github-a