



Univerzitet u Beogradu - Elektrotehnički fakultet

Katedra za signale i sisteme



## **DIPLOMSKI RAD**

# **PRAĆENJE PEŠAKA U VIDEO SEKVENCAMA POMOĆU FASTER R-CNN DETEKTORA I METODE NAJBЛИЖИХ SUSEDА**

**Kandidat**

Nikola Mijailović, br. indeksa 2018/0191

**Mentor**

dr Predrag Tadić, docent

Beograd, *septembar* 2022. godine

## REZIME

Ovaj rad se bavi problemom praćenja objekata u video sekvencama. Konkretno, radi se o praćenju kretanja pešaka na video sekvencama. Snimci potiču od sigurnosnih kamera javnih prostora kao što su parkovi i pešačke zone. Problem je rešen korišćenjem detektora pod nazivom *Faster R-CNN* i metodom globalnih najbližih suseda.

U radu su predstavljeni teorijski koncepti navedenih delova. Objašnjena su poboljšanja koja donosi *Faster R-CNN* u odnosu na početne *R-CNN* modele. Mreža je korišćena za detektovanje ljudi na svakom frejmu sekvence i njen izlaz se bazira na koordinatama pravougaonika koji uokviruju detektovane ljude i verovatnoći za tačnost te detekcije. Izlaz mreže služi kao ulaz za algoritam praćenja objekata.

Zatim, objašnjen je jednostavniji algoritam za praćenje samo jednog objekta i prikazan je njegov rad na generisanim podacima. Kao nadogradnja na ovaj model, implementiran je napredniji algoritam za praćenje kretanja više objekata istovremeno koji se bazira na metodi najbližih suseda. Prikazan je rad ovog algoritma, prvo, na generisanim podacima, gde se može zaključiti valjanost rada. Konačno prikazan je rad celog sistema na realnim podacima sa video sekvenci. Komentarisani su dobri primeri rada algoritma kao i situacije u kojima se dešavaju greške. Odgovoriće se na pitanje mogućnosti poboljšanja sistema i brzine rada celog sistema.

## ZAHVALNICA

Posebno se zahvaljujem mentoru dr Predragu Tadiću na izdvojenom vremenu i ukazanim smernicama koji su doprinele izradi ovog rada. Takođe, zahvaljujem se svojoj porodici na svojoj podršci tokom mog studiranja.

Nikola Mijailović

U Beogradu, *septembar* 2022.

## SADRŽAJ

REZIME.....	2
ZAHVALNICA .....	3
SADRŽAJ.....	4
1 UVOD .....	5
2 METODOLOGIJA RADA .....	6
2.1 Faster R-CNN.....	6
2.2 Praćenje objekata .....	8
2.3. Praćenje više objekata .....	10
3 REZULTATI I DISKUSIJA .....	14
5 ZAKLJUČAK.....	18
6 LITERATURA.....	19
PRILOG A.....	20

## 1 UVOD

Praćenje objekata na video sekvencama se sastoji iz dva dela. To su detekcija pešaka na snimcima, a zatim povezivanje detekcija nastalih od istih pešaka iz susednih frejmova video sekvence. Praćenje pešaka je veština koja je neophodna za rad samostalnih vozila.

Za same detekcije pešaka korišćena je mreža *Faster R-CNN*. U radu će biti prikazana njena arhitektura koja je preuzeta iz rada [1]. Dodata je funkcionalnost rada na celim video sekvencama. Iako se ova mreža može iskoristiti i za prepoznavanje mnogih drugih objekata, ovde je iskorišćena samo za prepoznavanje ljudi. Mreža je iskorišćena kao već istreniran model i kao takva je primenjena bez promena.

Drugi deo rada se ogleda u implementaciji algoritma za praćenje više meta na slici, tj. u ravni. Algoritam koji je iskorišćen nad detekcijama pešaka je *GNN* (eng. *Global Nearest Neighbor*) koji je implementiran tako da se može izboriti sa dosta slučajeva propuštenih i lažnih detekcija koje pravi gore pomenuta mreža. Filtar koji se pokazao kao odgovarajući uprkos tome što je najjednostavniji iz svoje grupe je Kalmanov filtar. O ostalim delovima i samom načinu rada GNN biće reči kasnije. Osim dva pomenuta problema, bitno je da se pojava ukrštanja detekcija svede na minimum. Ovaj algoritam ima mnogo širu primenu, može se primeniti nad podacima bilo kog senzora za praćenje objekata kao što su radari.

Cilj ovog rada je spajanje funkcionalnosti ova dva dela u jedan, tj. da se na originalnom snimku obeleže detekcija pešaka, gde je svakoj detekciji pridružen jedinstveni broj te detekcije id i gde je svaka detekcija obojena različitom bojom. Ovakav sistem, uz dodatna poboljšanja, mogao da se koristi kao pomoć bezbednosnim službama u vidu praćenje kretanja i ukupnog broja ljudi ne samo na ulici, već i u bilo kom zatvorenom prostoru.

U poglavlju METODOLOGIJA RADA biće objašnjen način rada i teorijske osnove algoritama. U poglavlju REZULTATI I DISKUSIJA biće prikazani i razmatrani rezultati i brzina rada. Biće pokazani primeri dobrog rada sistema, kao i karakteristične greške koje se javljaju prilikom rada. U ZAKLJUČKU će biti prokomentaran rad celog sistema i dat predlog za rešenje otkrivenih problema u radu.

## 2 METODOLOGIJA RADA

U ovom poglavlju će biti objašnjeni teorijski koncepti iz korišćenih algoritama za praćenje. Ali će prvo biti reči o samim detektorima tipa *R-CNN* i njihovim nadogradnjama.

### 2.1 Faster R-CNN

Faster R-CNN je drugo poboljšanje jednostavnijeg tipa neuralnih mreža R-CNN (eng. *Region-based Convolutional Neural Networks*). Ovo poboljšanje u odnosu na *Fast R-CNN* se ogleda u dosta bržem radu.

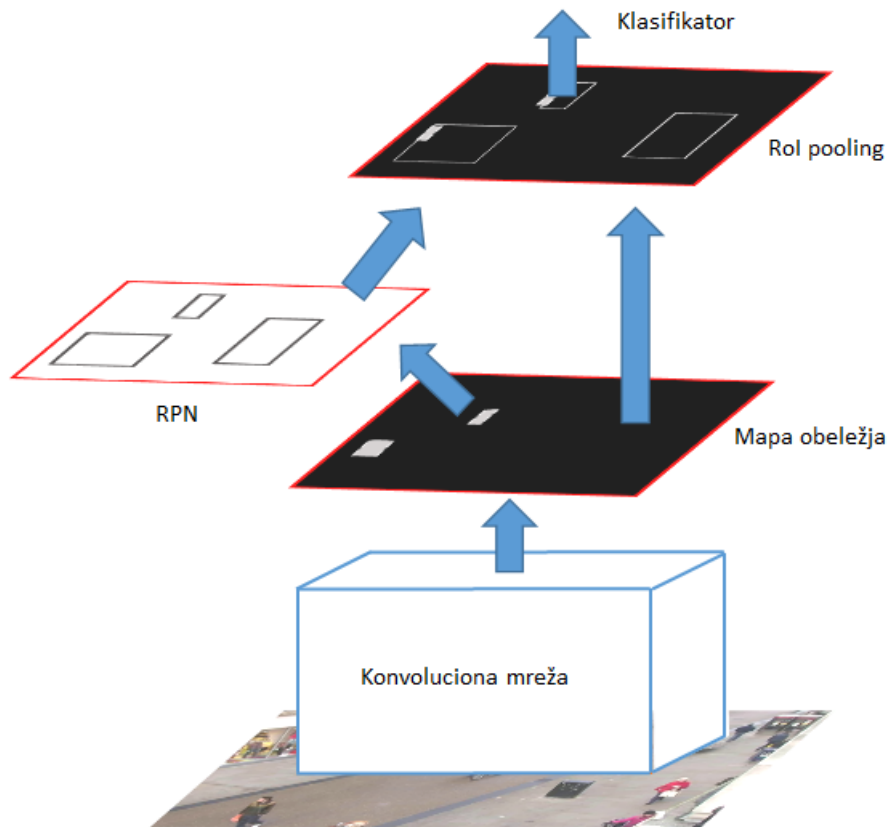
Konvolucione mreže mogu biti različitih arhitektura i dubina, one su specijalan tip veštačkih neuralnih mreža koje se obavezno koriste konvolucijom u barem jednom od svojih slojeva. Osim konvolucionih slojeva koji imaju ulogu da smanje broj parametara za učenje dozvoljavajući da mreže budu dublje, često se koriste *pooling* slojevi i potpuno povezani slojevi. *Pooling* sloj služi da smanje dimenzionalnost podataka iz jednog u sledeći sloj. U tom sloju se vrši združivanje izlaza nekih neurona najčešće koristeći operator maksimuma ili srednje vrednosti. Potpuno povezani slojevi se po pravilu nalaze na kraju konvolucionih mreža i služe za klasifikaciju slike. Bitna razlika koju unose konvolucione mreže je da neuroni dobiju kao ulaz samo informacije sa nekog broj neurona iz prethodnog sloja.

*R-CNN* je mreža koja pomoću trenirane konvolucione mreže izvlači obeležja iz slike. Prvo pomoću *Selective Search* algoritma se generiše oko 2000 regiona iz originalne slike. Ovaj algoritam za regione je deterministički, nije u stanju da „uči“. Zatim se svi regioni reskaliraju na predefinisane veličine i *CNN* iz njih izvlači vektor obeležja dužine 4096. Kao poslednji korak, vektor obeležja prolazi kroz *SVM* (eng. *Support Vector Machine*) koji klasifikuje region kao pozadinu ili kao neki od objekata. Mane obične *R-CNN* tokom njenog korišćenja se ogledaju u dugačkom vremenu izvršavanja. To se dešava zbog jako sporog algoritma za generisanje regiona, kao i zbog činjenice da se svaki region zasebno provlači kroz konvolucionu deo mreže.

Sledeće poboljšanje, pod nazivom *Fast R-CNN*, kao što stoji u imenu, donosi veću brzinu. Ova mreža ima dodatan sloj pod nazivom *RoI Pooling*, koji izvlači vektore obeležja jednakih dužina iz svih regiona jedne slike. Cela slika prolazi kroz duboku konvolucionu mrežu samo jednom, a zatim dobijena mapa obeležja prolazi kroz *RoI Pooling* sloj. Nakon toga dobijeni vektor obeležja prolazi kroz dve grane: granu za određivanje vrste objekta i za određivanje koordinata okvira kojim se objekti uokviruju. Za razliku od *R-CNN* koja se sastoji iz više odvojenih delova, ovde imamo spojene komponente u jednu. Osim brzine izvršavanja,

povećana je i preciznost. Ipak, koristi se isti algoritam za generisanje regiona kao kod *R-CNN*. To je ono što je u sledećem poboljšanju rešeno.

*Faster R-CNN*, naravno, još brža, uvodi mrežu za izdvajanje regiona sa slike umesto do tada korišćenog *Selective Search* algoritma. U ovom radu će biti korišćenja baš ona, pošto je od velikog značaja da se detekcija može izvršavati brzinom dovoljno velikom za primenu u realnom vremenu. Kod *Faster R-CNN* ulazna slika se šalje na konvolucionu mrežu, dalje dobijena mapa obeležja ide na novo-uvodenu mrežu za određivanje regiona od interesa RPN (eng. *Region Proposal Network*). Sada se predlozi regiona od interesa mogu prilagoditi specifičnim osobinama skupa na kom se mreža trenira. Dalje se nastavlja po istom principu kao kod *Fast R-CNN*. RPN kao ulaz uzima mapu obeležja iz poslednjeg konvolucionog sloja, tako da se RPN može spojiti sa ostatkom mreže i mogu se zajedno trenirati.



Slika 1 Prikaz elemenata *Faster R-CNN* [1]

Konvolucionni deo ove mreže je arhitekture tipa *VGG16*, a cela mreža je preuzeta kao trenirana na podacima *PASCAL VOC 2007*. Implementiran je rad mreže za celu video sekvencu.

Informacije od interesa su koordinate centralnog piksela, širine i visine detektovane oblasti, kao i sigurnost klasifikatora u tu detekciju.

## 2.2 Praćenje objekata

Drugi deo ovog rada bavi se algoritmima za praćenje detekcija i stvaranje putanji. U ovoj analizi započeto je od jednostavnijih primera gde postoji samo jedan objekat za praćenje. Merenja su generisana simulacijom. Korišćen je jednostavniji oblik sistema za praćenje u odnosu na onaj koji se koristi kasnije za praćenje više objekata na video sekvencama.

U nastavku je objašnjeno kako su merenja generisana. Osim merenja koja dolaze od tačne pozicije objekta ponekad se sa verovatnoćom  $P_D$  gubila detekcija, a u svakoj iteraciji postoji mogućnost za pojavu lažnih detekcija. Broj lažnih detekcija u svakoj iteraciji je generisan Laplasovom raspodelom, a njihova pozicija u prostoru uniformnom raspodelom.

Posmatranjem kretanja pešaka na ulici utvrđeno je da se dosta njihovog kretanja može modelovati pravolinijskim segmentima. Ipak, osim pravih delova, naravno, postoje i krivolinijski delovi koje se mogu modelovati delovima kružnice. S' obzirom na to, kao početni slučaj u kom je potrebno estimirati putanju samo jednog objekta, odlučeno je da se njegova putanja sastoji iz dva pravolinijska dela povezana polukružnom skretanju.

Kod ovde korišćenih algoritama za praćenje bilo je neophodno unapred pretpostaviti model kretanja objekta. Usvojen je model pravolinijskog kretanja sa konstantnom brzinom (eng. *Constant Velocity model*- CV). U opštem obliku model je dat kao linearni model:

$$x_{t+1} = Fx_t + Gw_t, \quad cov(w_t) = Q$$

$$y_t = Hx_t + e_t, \quad cov(e_t) = R$$

Stanje objekta je modelovano vektorom  $x_t = [p_x \ p_y \ v_x \ v_y]^T$ , gde su  $p_x$  i  $p_y$  koordinate objekta, a  $v_x$  i  $v_y$  brzine po odgovarajućim osama. Merenja su data vektorom  $y_t$  i ona su data kao očitavanje koordinata  $(p_x, p_y)$ . Matrice modela  $F, G, H$  su nepromenljive, tj. ne zavise od vremena. Matrica  $R$  predstavlja kovarijansu šuma merenja, ona ukazuje na nesigurnost koja dolazi sa senzora. Matrica  $Q$  je kovarijansa šuma u samom modelu, ona ukazuje na odstupanje u ponašanju od pretpostavljenog modela kretanja.

Za praćenje je upotrebljen Kalmanov filter, zbog svoje jednostavnosti i brzine rada. Kalmanov filter pravi predikcije za stanje objekta u sledećem trenutku prvo na osnovu estimacije iz prethodnog trenutka, a onda, po dolasku merenja vrši ispravku. Predikcija se vrši



na osnovu modela kretanja. U prvom koraku vrši se predikcija sledećeg stanja i računa se matrica koja predstavlja nesigurnost procene.

$$\begin{aligned}\hat{x}_{t|t-1} &= F\hat{x}_{t-1|t-1} \\ P_{t|t-1} &= FP_{t-1|t-1}F^T + GQG^T\end{aligned}$$

U drugom koraku se vrši popravka estimacije na osnovu pristiglog novog merenja iz trenutka  $t$ .

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - H\hat{x}_{t|t-1}), \quad P_{t|t} = (I - K_tH)P_{t|t-1},$$

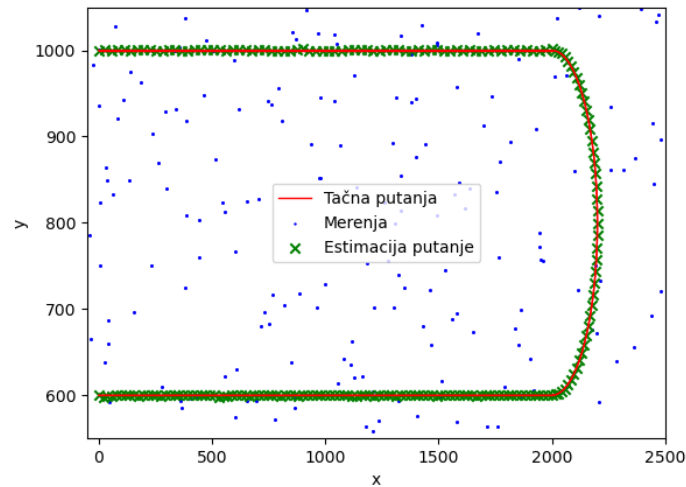
gde je Kalmanovo pojačanje  $K_t = P_{t|t-1}HS_t^{-1}$ , a  $S_t = HP_{t|t-1}H^T + R$ .

Za praćenje jednog objekta je potreban algoritam koji je zasnovan na Kalmanovom filtru, ali sa dodatnim obrađivanjem propuštenih i lažnih detekcija. Kako je za rad Kalmanovog filtra potrebno jedno merenje u svakoj iteraciji, jako je bitno da u slučaju kada se javlja više merenja izabere baš ono koje dolazi od tačne pozicije objekta. To se postiže prvo potpunim odbacivanjem merenja koja su daleko od estimacije pozicije objekta, a zatim se bira baš ono merenje koje je najbliže. Odbacivanje neodgovarajućih merenja (eng. *Gating*) se vrši na osnovu Mahalonobisove distance:

$$(y_t - \hat{y}_{t|t-1})^T S_t^{-1} (y_t - \hat{y}_{t|t-1}) \leq \gamma$$

Za prag se usvaja  $\gamma = 4.7$  što odgovara verovatnoći  $P_G = 0.9$ . Sada kada ostane manje merenja, ostaje još da se od preostalih izabere ono najbolje, tj. ono koje najbolje odgovara estimaciji na osnovu modela. Taj korak predstavlja povezivanje merenja sa objektom (eng. *Association*) i vrši se jednostavnim utvrđivanjem rastojanja estimacije od svih preostalih merenja. Ovakav način za izbor najboljeg merenja naziva se metodom najbližeg komšije (eng. *Nearest neighbor*). Kao svoj izlaz ova metoda daje konačno jedno merenje koje se dalje koristi u nastavku Kalmanovog filtra. U slučaju da posle odbacivanja neodgovarajućih merenja ne preostane ni jedno merenje, estimacija stanja u Kalmanovom filtru se ne može dodatno poboljšati, tj. vrši se promena estimacije samo na osnovu modela kretanja.

Na slici 2 dat je prikaz rada ovog algoritma na generisanoj putanji i merenjima. Može se uočiti da iako objekat naglo menja svoje kretanje i odstupa od pretpostavljenog modela kretanja, algoritam uspeva da održi tačnu putanju. Ovde je bilo potrebno podesiti matricu  $Q$  tako da se to omogući. Nad ovim primerom se može utvrditi dobra robusnost ovog osnovnog algoritma za praćenje.



Slika 2 Prikaz merenja i estimirane putanje

### 2.3. Praćenje više objekata

Sada se ovaj jednostavniji algoritam za praćenje jednog objekta može proširiti tako da prati više objekata istovremeno (eng. *Multi Target Tracking*). Biće primenjen algoritam globalno najbližih komšija (eng. *Global Nearest Neighbor - GNN*). Kada u svakom trenutku potencijalno postoji više objekata koje treba pratiti, jako je bitno da se merenja povežu sa odgovarajućim objektima, kao i da se lažne detekcije odbace. To se radi sličnim algoritmom za odbacivanje merenja kao pre, a povezivanje je dodatno unapređeno.

Uvodi se pojam putanje. Putanja može biti: probna, potvrđena ili izbrisana. Svaka putanja po njenom stvaranju je probna. Ako je utvrđeno da je probna putanja dovoljno sigurna, ona prelazi u potvrđenu. Potvrđena putanja može postati neaktivna, i onda dobija status izbrisane. Stvaranje nove putanje je inicirano time što se merenje ne može povezati ni sa jednom trenutno potvrđenom putanjom. Kako bi se otkrila valjanost tj. sigurnost neke putanje neophodno je definisati metriku  $L_t$ .

$$L_t = \log \frac{P(H_1|Y_t)}{P(H_0|Y_t)},$$

gde  $H_0$  predstavlja hipotezu u kojoj su sva merenje  $Y_t$  lažne detekcije, a  $H_1$  hipotezu u kojoj su merenja nastala od pravog objekta koji se prati.  $Y_t$  predstavlja samo merenja koja su prošla prethodno odbacivanje (eng. *Gating*). Metrika putanje se može računati rekursivno:

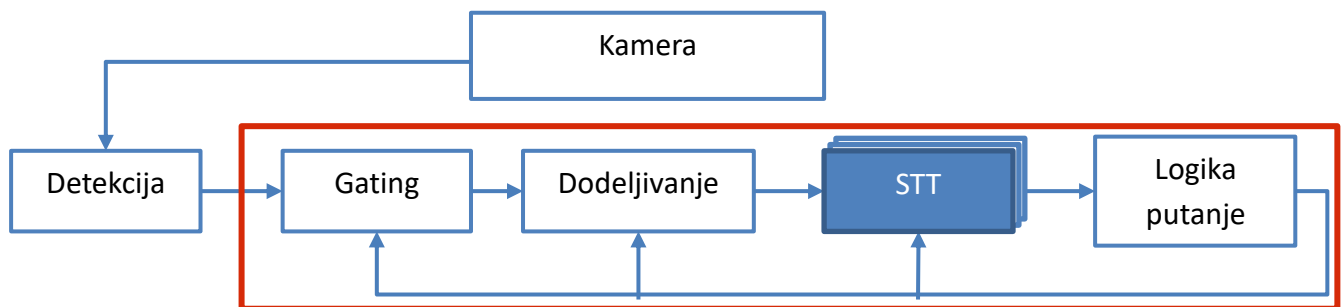
$$L_t = \lambda L_{t-1} + l_t,$$

gde je  $L_t$  doprinos u trenutku  $t$ , a  $\lambda$  predstavlja faktor zaboravljanja koji se dodaje radi uklanjanja efekta navijanja integratora. U [2] pokazano je da se  $L_t$  može računati na sledeća dva načina:

- $L_t = \lambda L_{t-1} + \log \frac{P_D P_G p(y_t | Y_{t-1})}{\beta_{FA}}$ , kada je dostupno  $y_t$
- $L_t = \lambda L_{t-1} + \log(1 - P_D P_G)$ , kada nema dostupnog merenja

Dodatno  $P_G$  je verovatnoća da merenje koje odgovara objektu nije odbačeno i usvaja se da je  $P_G = 1$ . Proizvod  $P_D P_G$  je verovatnoća da je odgovarajuće merenje detektovano i da je prošlo odbacivanje. Parametar  $\beta_{FA}$  je frekvencija pojavljivanja lažnih detekcija. Pri inicijalizaciji putanje  $L_t = 0$ .

Sada kad su objašnjene sve komponente sistema za praćenje pešaka dat je njihov grafički prikaz na slici 3. Kamera služi kao senzor koji prikuplja podatke. Zatim se pomoću *Faster R-CNN* koja vrše detekcije pešaka, i onda sledi algoritam za praćenje koji je uokviren crven bojom na slici 3.



Slika 3 Grafički prikaz sistema za praćenje objekata (eng. STT- Single Target Tracking)

Kao što je napomenuto, pošto sada postoji više merenja i putanja koje treba pratiti, bitno je da se definiše način njihovog uparivanja, odnosno način na koji će se merenje dodeliti putanji od koje pripada. Definiše se validaciona matrica  $V$ , gde svaki njen element  $v_{ij}$  ukazuje da li  $i$ -to merenje može da odgovara  $j$ -toj aktivnoj putanji. To se utvrđuje na isti način kao kod odbacivanja merenje za praćenje jednog objekta. Svako merenje se može:

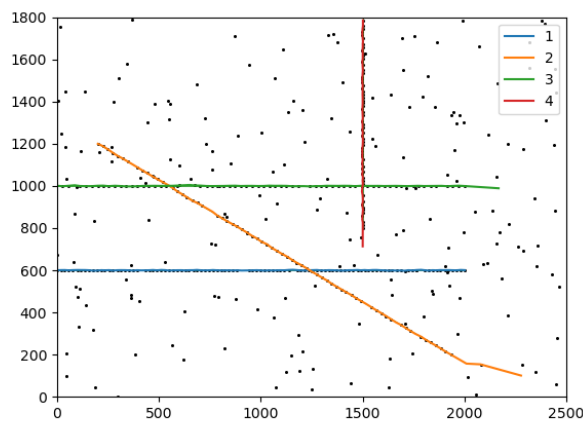
- iskoristiti za nastavak već postojaće neizbrisane putanje,
- proglašiti lažnom detekcijom ili
- poslužiti za nastanak nove putanje.

Takođe, definiše se i matrica dodeljivanja (eng. *Assignment Matrix*)  $A$ . Određeni elementi  $a_{ij}$  predstavljaju „cenu“ da se  $i$ -to merenje dodeli  $j$ -toj putanji, dok ostali ukazuju na mogućnost da se merenje proglašiti lažnom detekcijom ili da se iskoristi za novu putanju. Pre ovoga se iz

validacione matrice mogu uvideti nemogući parovi, što smanjuje broj elemenata za račun unutar matrice  $A$ . Vrednosti elemenata matrice  $A$  koji ukazuju na dodeljivanje  $i$ -tog merenja  $j$ -toj putanji je  $a_{ij} = \log \frac{P_D p_{t|t-1}^{(j)}(y_t^{(i)})}{1 - P_D P_G}$ , gde je  $p_{t|t-1}^{(j)}(\cdot)$  pretpostavljena funkcija gustine verovatnoće  $j$ -te putanje koja je u slučaju Kalmanovog filtra jednaka Normalnoj raspodeli  $N(y; \hat{y}_{t|t-1}^{(j)}, S_{t|t-1}^{(j)})$ .

Već je naglašeno da se svako merenje mora iskoristiti. Najbolji način za dodeljivanje se svodi na poznati optimizacioni problem dodele. Naime, iz svake vrste matrice  $A$  bira se tačno jedan element, dok se najviše jedan element bira iz svake kolone. Za rešavanje ovog problema iskorišćen je Jonker-Volgenant algoritam. Kao već implementirano rešenje koje koristi ovaj algoritam uzeta je metoda iz modula *scipy.optimize* pod nazivom *linear\_sum\_assignment* [7].

Zarad provere rada ovog algoritma i odabira parametara za naprednije algortime generisana su kretanja više objekata istovremeno. Dodate su lažne detekcije, a nekada su prava merenja bila i propuštena. Generisana su pravolinijska kretanja četiri objekta koja se kreću različitim brzinama. Na slici 4 dat je prikaz svih estimiranih putanja, gde tačke predstavljaju sva merenja.

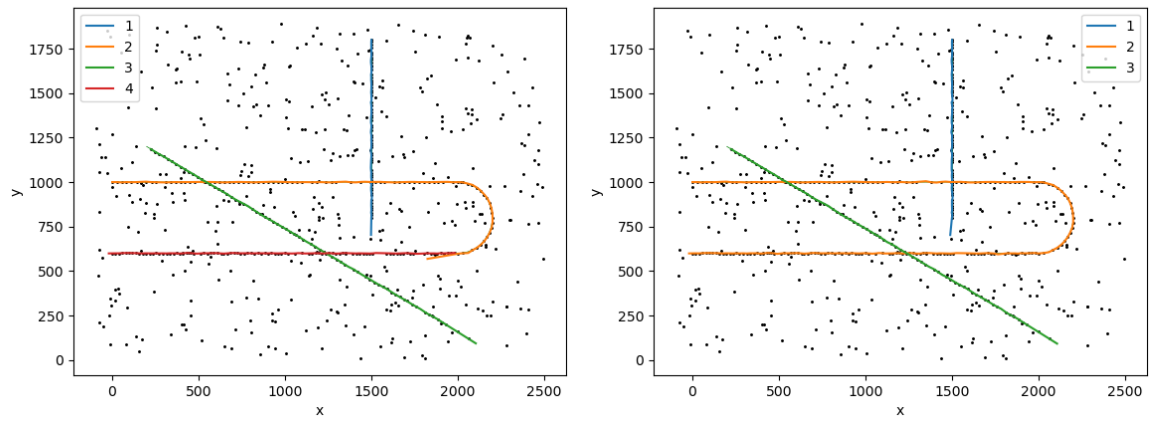


Slika 4 Prikaz merenja i estimiranih putanja

Sa slike 4 se može ustanoviti da u slučaju lepo razdvojenih putanja, algoritam bez problema uspeva da uhvati njihove tačne putanje.

Na slici 5 se javlja putanja koja ne odgovara pretpostavki o modelu kretanja sa konstantnom brzinom. Tu se može primetiti bitan uticaj matrice  $Q$ , kojom se zadaje varijansa procesnog šuma, na valjanost putanji. Dešava se da za postavljene manje vrednosti dijagonalnih elemenata  $Q$  algoritam ne uspeva da prepozna celu putanju, već je prekida na delu kada kretanje počne jako da odstupa od pretpostavljenog modela kretanja. Tada se na

tom mestu inicira nova putanja koja uspeva da otprati objekat do kraja. Vidi se da se praćenje kretanja objekata koji prati zadati model lepo estimira u oba slučaja.



Slika 5 Prikaz merenja i estimiranih putanja. Na grafiku levo matrica  $Q$  je manja deset puta

### 3 REZULTATI I DISKUSIJA

U ovom poglavlju će biti prikazani rezultati na video sekvencama dostupnim u [3]. Ovaj izvor sadrži dosta snimaka sa raznih sigurnosnih kamera, od kojih ću ja iskoristiti dva, ali i mnoge druge. Biće prikazan rad celog sistema od detektovanja mreže sve do konačno dobijenog snimka gde su različitim bojama i ID označene detekcije koje pripadaju različitim putanjama.

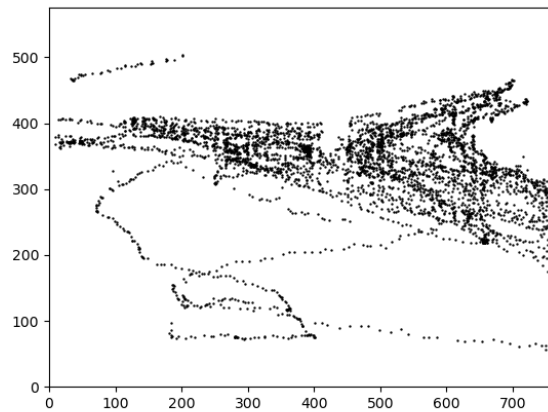
Skripte za pokretanje mreže napisane su u programskom jeziku *Python*. One su pokretane na servisu pod nazivom *Google Colaboratory*, koji pruža određenu količinu RAM memorije i korišćenje grafičkih kartica raznih proizvođača. Izlaz mreže se čuva u eksterni fajl, gde svaki red poseduje informacije o tačno jednoj detekciji i rednom broju frejma u kojem se ona javlja. Kao ulaz za algoritam praćenja, *GNN*, korišćeni su centri detektovanih oblasti sa frejmova iz video sekvence. Ovaj algoritam je pokretan lokalno na računaru u *Spyder* okruženju.

Kao prva ulazna sekvenca uzet je snimak sa ukupno 794 frejma. Broj frejmova po sekundi  $FPS = 7$ . Dat su primeri dva frejma posle izvršenih detekcija na slici 6. Slika 6 predstavlja izlaz *Faster R-CNN*. Ova dva frejma ukazuju na neke problema koje se javljaju pri detekcijama. Levo na slici 6 se može primetiti propuštena detekcija čoveka koji se nalazi iza bandere. Na frejmu desno se dešava lažna detekcija i to da su dve osobe koje idu zajedno spojene u jednu detekciju. Ipak, postoji i dosta dobro detektovanih primera, primećuje se, na frejmu levo, da se osoba u daljini lepo detektuje.



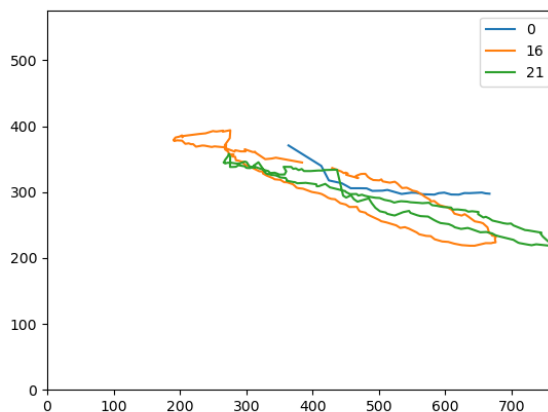
Slika 6 Prikaz dva frejma sa izlazne video sekvence *Faster R-CNN*

Na slici 7 je jednom tačkom prikazan geometrijski centar svake detekcije. Postoji dosta ukrštanja putanja. Problem koji se često dešava u radu ovog sistema je međusobna zamena id između detekcija koje se preklapaju.



Slika 7 Prikaz svih detekcija tokom cele video sekvence

U nastavku su prikazane neke od potvrđenih putanja tokom vremena. Na slici 8 se primećuje kako se putanja detekcije sa id 16 gubi.



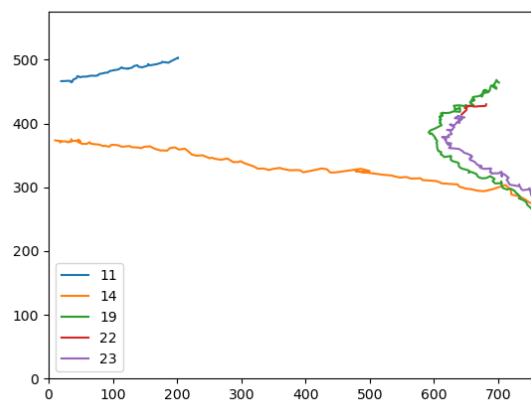
Slika 8 Estimacija putanja nekih detekcija

Na slici 9 je dato objašnjenje za gubitak putanje 16. Naime usled greške u detekciji, dve različite detekcije 16 i 21 su spojene i na dalje opstaje samo detekcija sa id 21. Zanimljivo je videti da je detekcija 16 iako se dosta nepravilno kretala, ipak neko vreme lepo praćena čak i po prolasku iza bandere gde je algoritam u stanju da lepo nastavi njenu putanju.



Slika 9 Prikaz praćenja nekih detekcija

Na slici 10 se mogu videti neki drugi primeri putanja. Putanja sa id 11 i 14 su u potpunosti tačne. Putanja detekcije 14 ukršta sa drugim putanjama što se može videti na slici 11. Putanja 19 je takođe lepo ispraćana, ali detekcija osobe pored nje se ponekad spajala sa detekcijom 19. Tako je, pogrešno, ista osoba dobila novi id. Njen id je prešao iz 22 u 23.



Slika 10 Prikaz putanji nekih detekcija

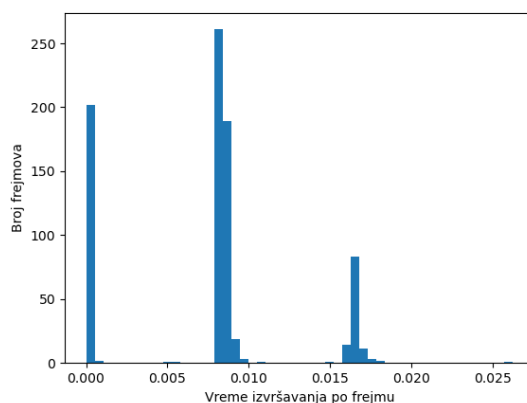
Na slici 11 je prikazana situacija u kojoj se *GNN* jako dobro izborio sa lošim detekcijama iz *Faster R-CNN*. Detekcije sa id 10 i 14 se ukrštaju. Sa prikaza konkretne situacije u nekoliko frejmova se može videti da se i jedna i druga putanja bez zamene vraćaju na svoje odgovarajuće detekcije po razdvajanju.





Slika 11 Prikaz praćenja detekcija sa id 10 i 14 u nekoliko frejmova

Vreme potrebno za detekcije sa strane *Faster R-CNN*, ne računajući početno učitavanje težina i ostalih podešavanja, je jednako oko 200ms po frejmu. Na slici 12 dat je histogram vremena izvršavanja *GNN* algoritma za svaki od frejmova. Prosečno je potrebno oko 7ms. Naravno, rad celog sistema bi trajao još duže zbog pravljenja novog snimka sa ucrtanim svim detekcijama.



Slika 12 Histogram vremena izvršavanja *GNN* algoritma po frejmu

## 5 ZAKLJUČAK

U rezultatima je prikazan rad celog sistema za praćenje pešaka na video sekvencama na osnovu detekcija generisanih *Faster R-CNN* modelom. Karakterističnim primerima je ukazano na najčešće greške koje sistem pravi. Veliki broj tih grešaka se dešava zbog nedovoljno preciznog rada detektora *Faster R-CNN*, iako je on dosta poboljšan u odnosu na običan model R-CNN. Takođe u prikazanim rezultatima za generisane slučajeve komentarisano je da treba pažljivo odabrati parametre modela kretanja tako da algoritam može ispratiti kretanje objekata. Model konstante brzine koji je korišćen u celom radu je izabran zbog brzine i jednostavnosti, ali postoji i drugi bolji modeli koji bi potencijalno doneli poboljšanje. Logika koja se može primeniti je da se izvrši kombinacija dva modela kretanja. Modelu koji se koristi tom idejom se nazivom *IMM* (eng. *Interacting Multiple Model*), kog njega se može zadati model konstantne brzine i recimo model konstantnog skretanja (eng. *Constant Turn Model - CT*). Takođe, umesto što se detekcije u *GNN* posmatraju samo kao tačke, moguće je posmatrati ih baš kao pravougaonike. Tada bi se omogućilo da se dodatnom proverom odnosa stranica pravougaonika koji definišu detekciju, spreče neke od zamena id.

Praćenje objekata na video sekvencama je tema koja je dosta napredovala. Sve više se javljaju aplikacije u kojima se praćenje može raditi u realnom vremenu. Takve aplikacije bi našle razne primene u bezbednosne svrhe. Moguće je ustanoviti neka sumnjiva kretanja koja bi bila zanimljiva takvim službama. Prikaz svih merenja sa snimka na slici 7 može poslužiti u analizi kretanja ljudi u nekom prostoru. To može naći i razne primene u projektovanju prostora u vidu statistika kretanja ljudi. Može se snimanjem nekog javnog prostora dobiti takva statistika koja će služiti da ukaže na mesta gde je najveća frekventnost ljudskih prolaza i da se na njima recimo postavi klupa ili neki drugi sadržaj.

## 6 LITERATURA

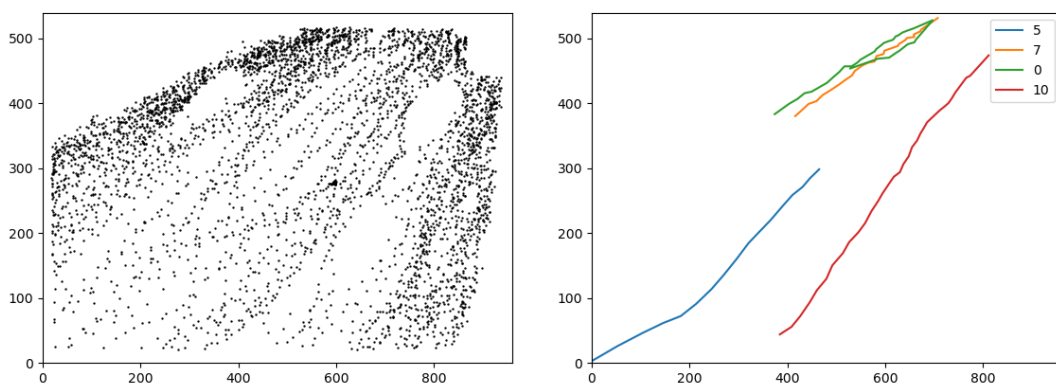
- [1] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, „Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks“, Januar 2016.  
Dostupno: <https://arxiv.org/pdf/1506.01497.pdf>
- [2] Gustaf Hendeby, Rickard Karlsson, „Course: Target Tracking“, 2021.  
Dostupno: <https://mtt.edu.hendeby.se/lecture.html>
- [3] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, Konrad Schindler, „MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking“, arXiv: 1504.01942, 2015.  
Dostupno: <https://motchallenge.net>
- [4] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, Ben Upcroft, „Simple online and realtime tracking“, 2016.  
Dostupno: <https://arxiv.org/pdf/1602.00763.pdf>
- [5] Ahmed Fawzy Gad, „Faster R-CNN Explained for Object Detection Tasks“, 2020.  
Dostupno: <https://blog.paperspace.com/faster-r-cnn-explained-object-detection/>
- [6] Diego Unzueta, „Kalman Filtering: A Simple Introduction“, Oktobar 2021.  
Dostupno: <https://towardsdatascience.com/kalman-filtering-a-simple-introduction-df9a84307add>
- [7] David F. Crouse, „On implementing 2D rectangular assignment algorithms“, IEEE Transactions on Aerospace and Electronic Systems, 52(4):1679-1696, Avgust 2016.

## PRILOG A

U prilogu su prikazani dodatni rezultati iz još jednog snimka na kojem je rad celog sistema testiran. Ovaj snimak je sa  $FPS = 2.5$ . Na slici 13 je prikaz detekcija sa dva frejma.



Slika 13 Dva frejma sa druge video sekvence



Slika 14 Prikaz svih merenja i nekih od potvrđenih putanja