

Projekt 5: Dolgoročno sledenje

Ema Leila Grošelj

I. UVOD

Pognala sem podan sledilnik na dveh različnih daljših sekvencah. Prilagodila sem ga tako, da bi postal dolgoročen.

II. PREIZKUSI

A. Kratkoročni sledilnik

Podan sledilnik sem pognala na dveh podanih daljših sekvencah: *dog* in *person20*:

dog

Precision: 0.20529101822884013
 Recall: 0.07609714740067852
 F-score: 0.11103566377246833

person20

Precision: 0.7358140023609
 Recall: 0.7358140023609
 F-score: 0.7358140023609001

Natančnost (precision) je $\frac{TP}{P}$, pokritost (recall) pa je $\frac{TP}{P}$. Nizka natančnost pri na primer *dog* pomeni, da je le malo od tistih, za katere je rekel, da so pozitivni tudi zares pozitivnih. Nizka pokritost, pa pomeni, da je spregledal večino resnično pozitivnih. F-ocena pa zajame pomen obeh metrik, in sicer predstavlja harmonično vsoto obeh (2-krat inverz vsote inverzov). Dejansko je glavna razlika med zaporedjema to, da *person20* nikoli ne izgine, prav tako se njegova podoba spreminja bolj počasi. Zato ima tudi toliko boljše rezultate.

B. Dolgoročni sledilnik

Izbrala sem si *dog*, ker je imel najmanjšo velikost datoteke, ampak sigurno ni bil najlažji. Problematično je že to, da na začetku ni veliko časa prosto videti, kar bi pripomoglo k temu, da bi si ga kratkoročni sledilnik dobro zapomnil. Sledijo pogoste okluzije z nogami, kar pomeni postopno učenje nog. Preden sem dodala vzorčenje v bližini, kjer je bil pes nazadnje zaznan, mi je sledilnik dobro ocenil tudi okolico in preskočil tja. Poleg tega je kljub temu, da je bil kandidat točno na psu, tudi zaradi druge barve ozadja sledilnik dal majhno oceno. Sledilnik sem malenkost izboljšala le pri pokritosti.

1) *Prag preklopa med načini*: Za **prenehanje sledenja** sem uporabila, kar oceno, ki jo vrne kratkoročni sledilnik. Prag sem pri psu nastavila na 3. Primerjava je zavedena pri 1 in 2.

Za **začetek sledenja** pa sem začela upoštevati, še oddaljenost kandidata od zadnje zaznane lokacije (poleg tega, da sem kandidate že izbirala normalno okoli te zadnje zaznane lokacije): $score * (1/dist) * \log(n_fails + 1) * 10$. Poleg tega se ocena s številom neuspešnih poskusov detekcije večja, da zagotovimo, da enkrat le preseže prag. Z 10 sem pomnožila, da sem dobila podobno skalo kot je ta pri oceni iz kratkoročnega sledilnika. Vendar se je na koncu izkazalo, da je bolje, da uporabim samo oceno, ki jo vrne kratkoročni sledilnik. Po primerjavi treh možnosti sem prag (start_treshold) nastavila na 3, saj je tam maksimalna F-ocena.

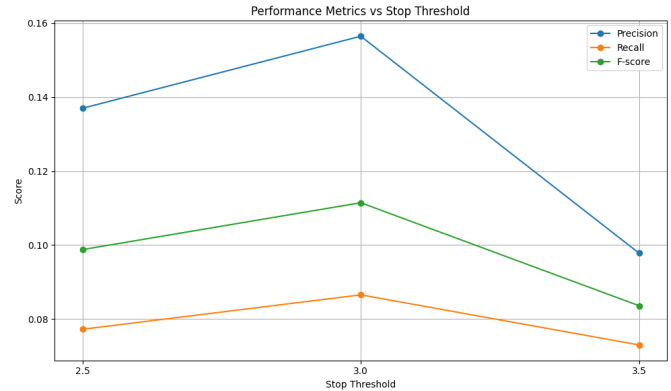


Figure 1: Prag ustavljanja.

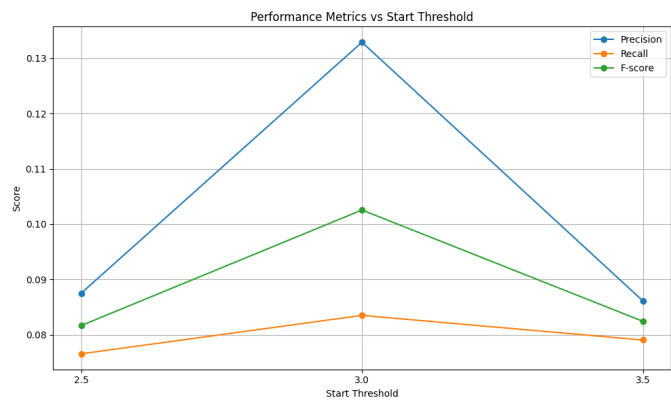


Figure 2: Prag začetka.

Optimalnost je relativna stvar. Mogoče bi v kakšni situaciji bilo bolje, da maksimiziram samo natančnost oz. samo pokritost. Poleg tega bi bilo dobro večkrat ponoviti preizkus za prav iste parametre, saj pri izbiri kandidatov nastopa najljubšost. Tako bi dobili občutek koliko variirajo rezultati in s tem preverili, koliko gre zaupati naši nastavitvi parametrov.

2) *Kandidati med redetekcije*: V fazi redetekcije v odvisnosti od parametra σ okoli zadnje zaznane lokacije namečem kandidate, pri čemer v odvisnosti od števila neuspešnih redetekcij povečujem σ . Poleg tega varianco prilagodim tudi velikosti tarče.

Uvedla sem parameter *lokalnost*, ki pomeni, koliko časa iščemo objekt v okolici, kjer smo ga zadnjič videli (sicer povečujočim σ). Ko tolikokrat ne uspemo, izbiramo kandidate enakomerno naključno. Da je *lokalnost* enaka nič torej ustreza temu, da kandidate skos generiramo enakomerno naključno. Na grafu 3 sem prikazala, kako spreminjanje *lokalnosti* vpliva na metrike; vidimo, da povečanje lokalnosti izboljša natančnost, vendar poslabša pokritost.

$$\Sigma = \sqrt{n_fails + 1} * (height + width) / 2 * \sigma * I_2$$

Pričakovala bi, da bo število kandidatov povečalo verjetnost uspeha, vendar tudi povečalo čas, potreben za računanje. Tako kot smo rekli na predavanjih je pri sledenju v realnem času

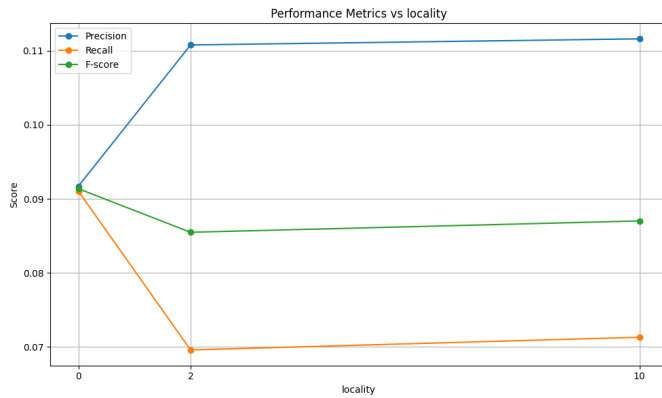


Figure 3: Lokalnost.

dovolj, da naredimo samo toliko kot lahko v času ene časovne enote, da ne pride do zakasnitev, poleg tega pa bi bili izračuni že zastareli. Meritve sem prikazala na 4, kjer se vidi, da dejansko najboljša izbira je več kandidatov. Na sliki 5 vidimo, da za redetekcijo rabimo manj korakov.

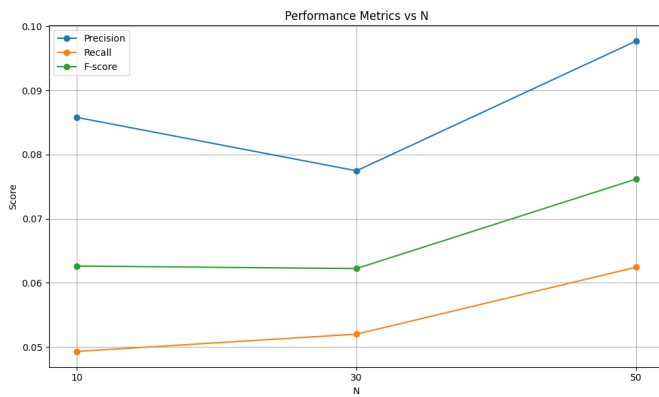


Figure 4: Metrike v odvisnosti od števila kandidatov.

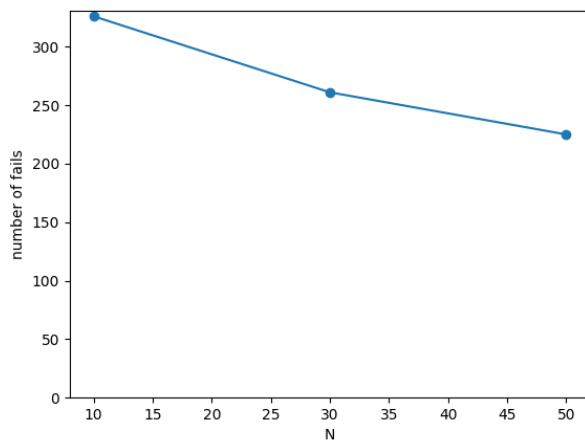


Figure 5: Število napak v odvisnosti od števila kandidatov.

Na slikah 6 in 7 sem prikazala kandidate in redetekcijo.



Figure 6: Redetekcija.



Figure 7: Sledenje.

III. ZAKLJUČEK

Sigurno bi lahko bolj izboljšala delovanje, morda bi to bilo tudi lažje na kakšnem manj zahtevnem zaporedju kot je bil *car9*. Se mi pa vseeno zdi *dog* trd oreh za tak kratkoročen sledilnik, kot smo ga dobili. Se mi zdi, da bi bilo dobro tudi "potvikati" njegove parametre, da bi ustrezali *dog - u*. Sama sem se nekoliko poigravala s *scale_step*.