

Projekt 5: Dolgoročno sledenje

Ema Leila Grošelj

I. UVOD

Pognala sem podan sledilnik na različnih daljših sekvencah. Prilagodila sem ga tako, da bi postal dolgoročen in ga pognala na *car9*.

II. PREIZKUSI

A. Kratkoročni sledilnik

Podan sledilnik sem pognala na podanih daljših sekvencah: *dog*, *car9* in *person20* (urejeni po padajoči težavnosti):

```
dog
Precision: 0.20529101822884013
Recall: 0.07609714740067852
F-score: 0.11103566377246833

car9
Precision: 0.6386934594221161
Recall: 0.27124927121812065
F-score: 0.38078250326398855

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 *person20* in *dog* to, da *person20* nikoli ne izgine, prav tako se njegova podoba spreminja bolj počasi. Zato ima tudi toliko boljše rezultate. *Car9* ima dokaj počasi spreminjajočo podobo in nobenih nenadnih premikov.

```
car9
(threshold_stop = 4, threshold_start = 3,
N = 30, locality = 100, sigma = 1)
Precision: 0.6712779630237834
Recall: 0.6580596103724369
F-score: 0.6646030679332485
```

B. Dolgoročni sledilnik

Sprva sem izbrala zaporedje *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.

Ker mi sledilnika na *dog* ni uspelo izboljšati sem se preusmerila na *car9*. Gre za zelo šolski primer, pri katerem zelo pomaga tudi vizualizacija, saj rabiš izpopolniti le reakcijo na eno "oviro". Tu je avto dosti časa na voljo za učenje njegove podobe.

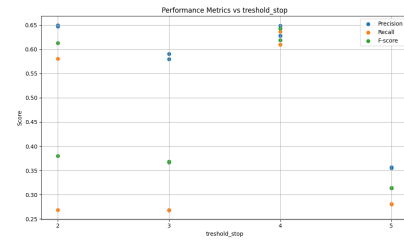


Figure 1: Prag ustavljanja.

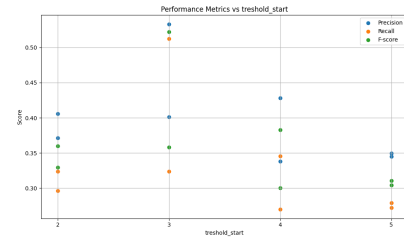


Figure 2: Prag začetka.

1) *Prag preklopa med načini*: Za prenehanje sledenja sem uporabila, kar oceno, ki jo vrne kratkoročni sledilnik. Za *threshold_stop* sem določila optimalno vrednost 4. Lokalna optimalnost izbire (če se ne spuščamo v decimalke) je prikazana na sliki 1.

Za nekaj časa sem imela v implementaciji tudi popravek ocene, ko smo v načinu redetekcije, v katerem sem upoštevala še oddaljenost kandidata od zadnje zaznane lokacije:

$$score * (1/dist) * \log(n_fails + 1) * 10.$$

Vendar se je pokazalo za dovolj, da oddaljenost upoštevam pri izbiri kandidatov, tako da sem popravek ocene opustila. Za *threshold_start* sem določila optimalno vrednost 3. Izbiro utemeljujem s sliko 2.

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

2) *Lokalnost in sigma*: 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 0 torej ustreza temu, da kandidate vseskozi generiramo enakomerno naključno. Na grafu 3 sem prikazala, kako spreminjanje *lokalnosti* vpliva na metrike. Moram povedati, da je bilo največje število zaporednih neuspehov nekje 40, zato so vse nastavitve *lokalnosti* na večje vrednosti od 40 ekvivalentne. Osebnostno sem se odločila, da bom *lokalnost* nastavila, kar na tako vrednost (100), da ne pride do enakomerne naključnosti, čeprav mogoče meritve na grafu malenkost preferirajo, *lokalnost* enako 10.

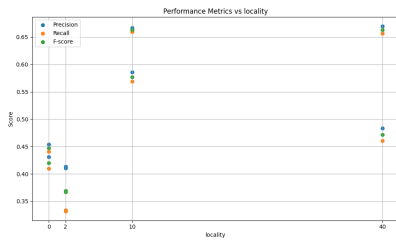


Figure 3: Lokálnost.

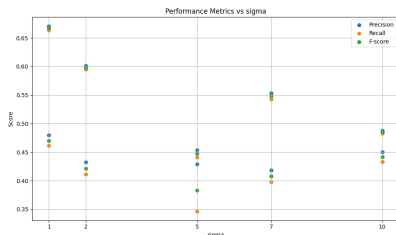


Figure 4: Parameter sigma.

V fazi redetekcije v odvisnosti od parametra σ okoli zadnje zaznane lokacije namečim kandidate, pri čemer v odvisnosti od števila neuspešnih redetekcij povečujem dejanski σ . Poleg tega dejanski σ prilagodim tudi velikosti tarče:

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

Meritve za različne parametre σ sem prikazala na grafu 4. Ker iz drugih meritev vem, da se z nastavitvijo σ na 10 ponavadi dobi rezultate nad 0.65, sem tudi tukaj skeptična, glede števila ponovitev meritev z istimi parametri. Ampak za najboljšo se je izkazala σ enaka 1.

3) *Število kandidatov med redetekcijo*: 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 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. Ko sem pognala sledilnik z različnimi N pri *lokalnosti* enaki 100 tj. centriranost okoli izhodišča s povečujočim σ , nisem dobila pričakovanih rezultatov. Prikazala sem jih na sliki 5. Meritve sem ponovila še za enakomerno naključne kandidate in dobila čisto drugačne rezultate. Oboje sem dvakrat ponovila in dobila isto. Morda gre za kakšno past, v katero pademo, če imamo preveč poskusov. Na primer, da nam sledilnik preskoči na ozadje. Tega nečemo. Izgleda, da več ni vedno tudi bolje.

Na sliki 6 vidimo, da za redetekcijo rabimo manj korakov, če povečamo število kandidatov.

Na sliki 7 sem prikazala kandidate in redetekcijo.

III. ZAKLJUČEK

Nekoliko sem uspela izboljšati vse metrike, vendar je vprašanje, koliko je bila to sreča in kolikšne so te metrike v povprečju. Pravtako sem obupala nad *dog*, ki je neprimerno bolj zahteven primerek. Kar bi še lahko poskusila pa nisem, je da bi za vsakega kandidata dodala med kandidate tudi njegovo manjšo in večjo verzijo in s tem zagotovila več možnosti za redetekcijo ob vmesni spremembi skale.

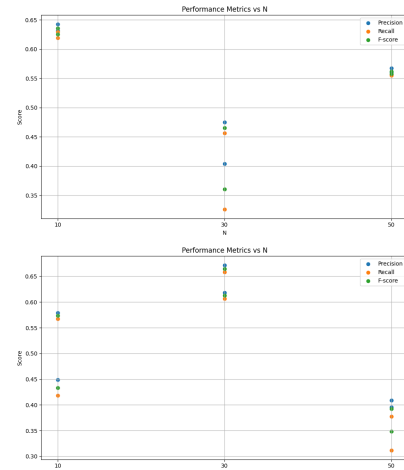


Figure 5: Število kandidatov s centriranostjo (zgoraj) in enakomerno naključnostjo (spodaj).

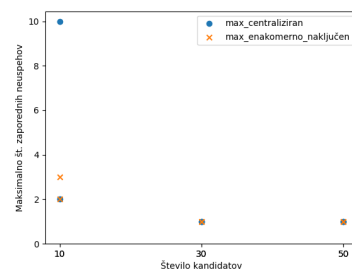


Figure 6: Maksimalno število zaporednih napak v odvisnosti od števila kandidatov.

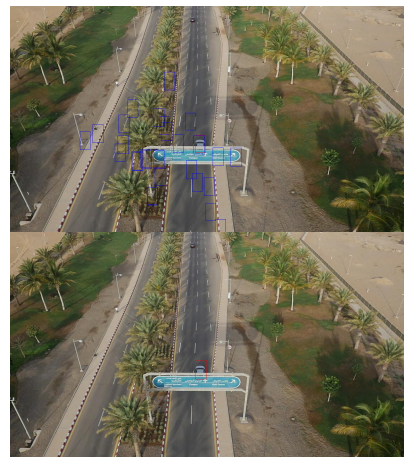


Figure 7: Redetekcija (zgoraj) in sledenje (spodaj).