Soft Kompjuting E2 – 2021/22

Nedeljni izazov #1 – Obrada slike
(Analiza zauzetosti parkinga putem nadzornih kamera)

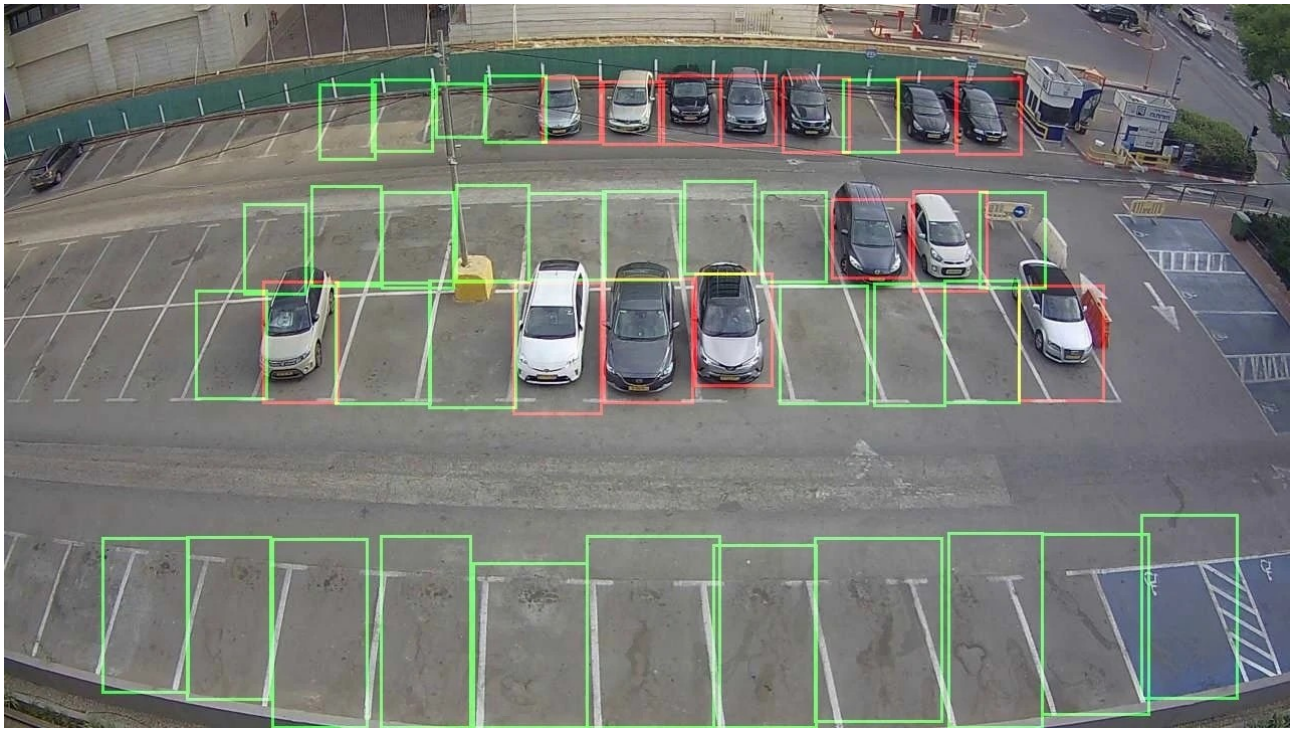
Motivacija

Život u 21. veku se ne može zamisliti bez automobila, kao glavnog prevoznog sredstva u urbanim i ruralnim sredinama. Međutim, kako potreba za automobilima raste, istovremeno raste i potreba za parking mestima u urbanim sredinama.

Zbog toga se u većim gradovima grade višespratne parking garaže sa video nazdorom. Ovaj način parkiranja je najbezbedniji i sa stanovišta očuvanja vozila od štete nastale od strane drugih ljudi, kao i sa stanovišta zaštite vozila od krađe.

Kamere se obično koriste samo za nadzor u slučaju nedozvoljenih radnji i pristup video sadržaju imaju samo ovlašćena lica i entitet koji upravlja garažom, a za druge namene obično nisu dovoljno iskorišćene.

Neke javne garaže koriste kamere i za evidentiranje vozila koja su ušla u samu garažu (najčešće čitanjem registarskih tablica), dok neke vode računa samo o proceni zauzetosti parkinga u nekom momentu. Ove informacije su u razvijenim zemljama dostupne i online, pa pre nego što se uputite ka nekoj javnoj garaži, možete prethodno proveriti da li u njoj uopšte postoji slobodnog mesta.



Slika 1: Primer sistema za brojanje slobodnih i zauzetih mesta na parkingu, putem sigurnosne kamere

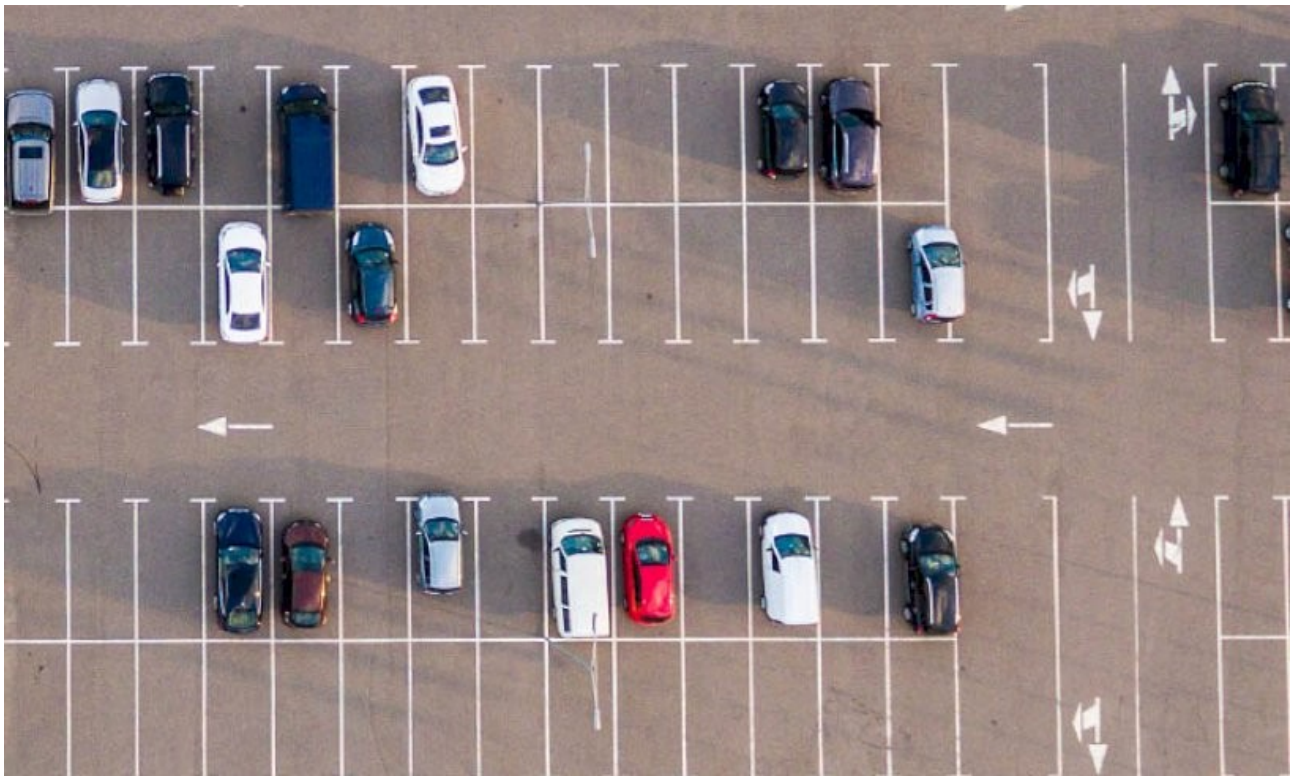
U ovom izazovu ćemo se baviti pomenutim zadatkom, i probaćemo da automatski brojimo koliko je automobila trenutno prisutno na parkingu, bez intervencije čoveka. Ukoliko neki javni parking koristi ovaj softverski sistem, brojanjem automobila mogu vrlo lako izračunati i broj slobodnih mesta (svaka garaža zna koliko parking mesta imaju dostupno) i te informacije automatski javno deliti.

Zadatak

Naš ovonedeljni zadatak će biti detekcija i brojanje automobila na fotografijama parkinga, **koje su pokupljene iz sigurnosnih kamera**. Brojanjem automobila se lako može doći do broja slobodnih mesta na parkingu (pretpostavljamo da vlasnik parkinga/korisnik ovog softverskog sistema zna koliko parking mesta je izgradio).

Pošto je u ovonedeljnom zadatku dozvoljeno koristiti samo tehnike računarske vizije kroz openCV biblioteku, fotografije su iz ptičje perspektive da bi se problem pojednostavio. U suprotnom bi se morali koristiti modeli veštačke inteligencije da bi se detektovali automobili iz različitih uglova i pod različitim uslovima (što ćemo mi i raditi na vežbama, ali tek u nastavku semestra).

Primer fotografije na kojoj je potrebno prebrojati automobile se nalazi ispod.



Izvršiti brojanje automobila korišćenjem tehnika računarske vizije kroz OpenCV biblioteku i programski jezik Python.

Neki osnovni koraci u rešavanju ovog izazova bi mogli biti:

1. segmentacija slike (binarizacija) - sami treba da odredite na koji način je to najbolje uraditi. Budite kreativni,
2. Uklanjanje šuma i post-procesiranje problematičnih situacija (možda spojenih kontura, možda razdvojenih kontura i sl), možda detekcija linija i sl.,
3. Brojanje automobila.

Na vežbama smo prošli sasvim dovoljno teorijskih i praktičnih osnova za rešavanje ovog izazova. ***Budite kreativni i primenite ih na svoj način, tako da dobijete što bolje rezultate.***

Format koda za ocenjivanje (isto za sve izazove)

Kod koji upload-ujete u GoogleDrive folder treba da zadovolji neke kriterijume da bi ga platforma za ocenjivanje analizirala na pravi način. Glavna ograničenja su sledeća:

1. **Fajl main.py se mora nalaziti u korenu vašeg foldera.** Ukoliko to nije slučaj, platforma neće biti u mogućnosti da pokrene vaše rešenje i nećete biti ocenjeni.

2. **Fajlove main.py i evaluate.py nije dozvoljeno menjati.** Ovi fajlovi su direktno korišćeni od strane platforme da bi ocenjivanje bilo moguće.

3. **Vaša implementacija treba da bude u fajlu process.py.** U ovom fajlu se nalazi neimplementirana metoda koja ima jasno naznačen ulaz i izlaz. Metoda je automatski uklopljena u ostatak koda (poziva se iz main.py) i nema potrebe da je ručno pozivate. **Vaš zadatak je da implementirate traženu metodu i da obezbedite da vraća ono što se od vas traži.**

4. **Dozvoljeno je kreiranje novih python fajlova, koje možete pozivati iz process.py.** Ukoliko želite da deo koda izdvojite u druge fajlove I da onda kroz python import koristite u process.py, to je dozvoljeno. Dok god poštujete sve prethodne korake, ne bi trebalo biti problema.

5. **U kodu koji okačite na platformu nemojte koristiti sistemske pauze i slične mehanizme koji zahtevaju reakciju korisnika, pošto u tom slučaju rešenje neće biti pokrenuto.**

Pokretanje rešenja i evaluacija

Da biste pokrenuli rešenje na svojoj mašini i proverili kolika je postignuta tačnost, potrebno je uraditi sledeće:

1. Implementirati metodu u **process.py** traženom logikom. Ovaj fajl **ne** pokrećete direktno.
2. Pokrenuti **main.py** (iz pycharm-a na Run, ili iz terminala komandom “python main.py” uz prethodno aktiviranje odgovarajućeg virtuelnog okruženja). Pokretanje main.py fajla će izgenerisati **result.csv** fajl, tako što će pozvati prethodno implementiranu metodu za sve primerke iz skupa podataka.
3. Pokrenuti **evaluate.py** fajl (iz pycharm-a na Run, ili iz terminala komandom “python evaluate.py” uz prethodno aktiviranje odgovarajućeg virtuelnog okruženja). Ovaj fajl će učitati result.csv koji je prethodno generisan i izračunati tačnost. Izlaz ovog fajla je samo broj koji pokazuje procenat tačnosti trenutnog rešenja.

Ocenjivanje (isto za sve izazove)

Ocenjivanje upload-ovanog koda će biti izvršavano iterativno, po sledećim pravilima:

1. Platforma će automatski vršiti download koda, jednom u 24h i vršiti ocenjivanje.

2. U toku jednog dana možete imati neograničen broj upload-a. Ocenjivanje će svakako biti pokrenuto samo jednom na kraju dana i biće ocenjen kod koji u tom trenutku bude u folderu na Google Drive-u.

3. Platforma vrši ocenjivanje za prethodni dan u periodu **od 3:00 iza ponoći do 8:00 ujutru narednog dana**, pa u tom periodu nije dozvoljeno menjanje fajlova.

4. Ukoliko izazov traje 7 dana, studenti tehnički imaju 7 pokušaja da reše izazov. Platforma će ocenjivati kod svaki dan. Na rang listu će se računati **najbolji rezultat** iz svih ciklusa ocenjivanja. Zbog toga je bolje da što ranije rešite izazov, pošto ćete imati više pokušaja da ispravite nešto i postignete još bolji rezultat. Ako bilo koji pokušaj bude detektovan kao plagijat, student dobija godinu dana zabrane polaganja.

5. Svaki dan će studenti dobijati izveštaj u formi txt fajla u svom Google Drive folderu. Ovo se odnosi samo na studente koji su postavili nešto u svoj folder. Izveštaj se generiše svaki dan, bez obzira na to da li ste šta menjali u folderu tog dana. Tako ćete na dnevnom nivou biti ažurirani činjenicom gde se nalazite na rang listi.

6. U izveštaju niko neće imati informaciju gde se tačno nalazi na rang listi. Dobićete informaciju da li se nalazite u TOP 5, TOP 10, TOP 25 ili TOP 50 studenata. Ako ste dobili informaciju da ste u TOP 25, to znači da se nalazite izmedju 11. i 25. pozicije i da možete poboljšati rešenje da popravite rang. Tačan rang će biti objavljen naknadno, tek na kraju izazova.

Dozvoljene biblioteke i podešavanje okruženja

U sklopu ovog izazova je dozvoljeno koristiti sledeće biblioteke uz Python 3.6:

- numpy
- scipy
- openCV verzija 3.4.1 (mada može i bilo koja verzija koja počinje sa 3.x.y)

Nije dozvoljeno korisiti druge biblioteke, kao ni korišćenje pretreniranih modela (kaskadnih klasifikatora, konvolutivnih neuronskih mreža i slično), pošto je poenta izazova savlađivanje napredne obrade slike korišćenjem samo OpenCV biblioteke.

Instaliranje:

Za kreiranje okruženja i instalaciju biblioteka je potrebno preuzeti najnoviju Anaconda distribuciju sa njihovog zvaničnog sajta i instalirati je. Anaconda postoji za sve moderne operativne sisteme. Nakon instaliranja možete preći na kreiranje virtuelnog okruženja i instaliranje biblioteka u njega.

Detaljniji opis šta virtuelna okruženja predstavljaju možete naći u sklopu **v0** na github repozitorijumu predmeta (https://github.com/ftn-ai-lab/sc-2021-e2/blob/master/v0-priprema/podesavanje-okruzenja.ipynb), i u video tutorial-u.

1. Kreirati virtuelno okruženje (iz terminala na Linux i MacOS, ili Anaconda prompt na Win)

   ```
   conda create -n soft-env python=3.6
   ```

2. Aktivirati okruženje (ukoliko ćete fajlove kasnije pokretati iz terminala, a ne iz PyCharm-a)

   ```
   source activate soft-env
   ```
   ili
   ```
   conda activate soft-env
   ```

3. Instalirati biblioteke

   ```
   conda install -n soft-env -c conda-forge opencv=3.4.1
   ```

4. Preuzeti i instalirati pyCharm Community razvojno okruženje, koje je preporuka za razvoj python projekata. Otvoriti projekat koji je deo ovog izazova.

5. Podesiti interpreter za projekat tako što ćete se povezati na python instancu iz prethodno kreiranog virtuelnog okruženja. Uputstvo je nalazi na kraju sledećeg fajla, koji je na github repozitorijumu predmeta (https://github.com/ftn-ai-lab/sc-2021-e2/blob/master/v0-priprema/podesavanje-okruzenja.ipynb)

6. Sve je spremno. Desni klik na odgovarajući fajl i onda Run ili Debug. Ukoliko importovanje cv2 biblioteke puca u skripti, proverite da li ste dobro instalirali openCV i da li ste dobro povezali projekat sa virtuelnim okruženjem.