

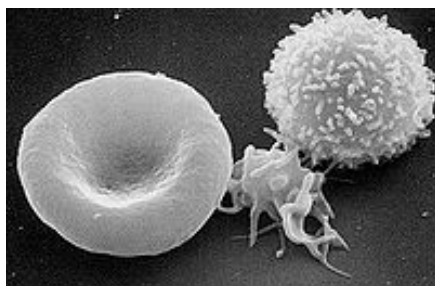
Soft Kompjuting E2 – 2020/21

Nedeljni izazov #1 – Obrada slike (Detekcija leukemije sa mikroskopskih snimaka razmaza krvi)

Motivacija

Krv čini oko 55% tečne krvne plazme i oko 45% čvrstih krvnih ćelija koje se nalaze u plazmi. Krvna plazma je bezbojna, odnosno žućkasta tečnost: vodeni rastvor organskih i neorganskih materija.

Krvne ćelije su crvena krvna zrnca (**eritrociti**), bela krvna zrnca (**leukociti**) i krvne pločice (**trombociti**). Crvena krvna zrnca daju krvi crvenu boju, sastavljena su od belančevina i holesterola, a sadrže i hemoglobin. Bela krvna zrnca su postojane ćelije koje se razmnožavaju deobom. Krvne pločice su najmanja krvna zrnca; oni su u uskoj vezi sa zgrušavanjem krvi.

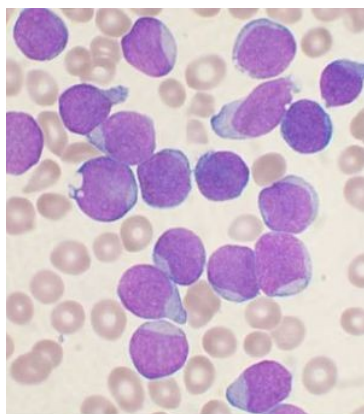


Ilustracija 1: Levo na desno - eritrocit, trombocit i leukocit

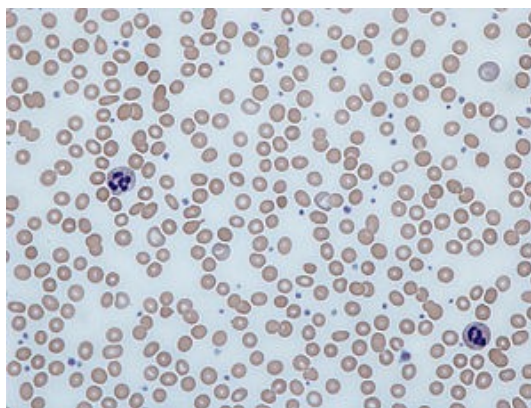
Odnos zapremine krvne plazme i krvnih ćelija kod zdravih osoba je **konstantan**. Zapremina krvnih ćelija kod muškaraca je nešto veća nego kod žena.

Loš odnos broja eritrocita, leukocita i trombocita može ukazivati na različite bolesti, a primeri nekih od njih su:

- **Anemija** - stanje nedostatka crvenih krvnih zrnaca. U tom slučaju, mali broj eritrocita (crvenih krvnih zrnaca) sadrži manje hemoglobina, zbog čega je dostava kiseonika organima kao što su mozak, srce i mišići smanjena i samim tim ćelije dobijaju manje hranjivih materija.
- **Leukemija** - U krvi i koštanoj srži obolelih od ove bolesti može se najčešće naći mnoštvo nezrelih leukocita (blasti).
- **Trombocitoza** - povećan broj krvnih pločica nosi rizik za nastanak tromboze (spontanog zgušnjavanja krvi)
- **Trombocitopenija** - smanjen broj krvnih pločica. Može uzrokovati spontana, tačkasta krvarenja koja se zovu trombocitne purpore. Ovakva krvarenja se mogu javiti svuda po telu i na sluzokoži



Ilustracija 2: Leukemija - mnoštvo nezrelih leukocita



Ilustracija 3: Anemija - crvena krvna zrnca su mnogobrojna, jako mala i svetla

Zadatak

Naš ovonedeljni zadatak će biti detekcija leukemije na osnovu broja crvenih i belih krvnih zrnaca iz fotografija mikroskopskih snimaka razmaza krvi.

U zavisnosti od vrste bolesti, koraci analize fotografija razmaza krvi bi bili različiti, kao na primer:

1. Detekcija krvnih zrnaca,
2. Prepoznavanje tipa krvnih zrnaca (koje krvno zrnce je u pitanju),
3. Računanje odnosa broja krvnih zrnaca, analiza boje prepoznatih krvnih zrnaca (za neke bolesti je ovo bitno) i slično,
4. Dijagnostika bolesti uz korišćenje informacija iz prethodnih koraka.



Ilustracija 4: Primer fotografije razmaza krvi na kojoj treba prebrojati crvena (RBC) i bela krvna zrnca (WBC)

Izvršiti brojanje crvenih krvnih zrnaca (RBC) i belih krvnih zrnaca (WBC) korišćenjem tehnika računarske vizije kroz OpenCV biblioteku i programski jezik Python, a nakon toga odrediti da li pacijent boluje od leukemije ili ne.

Neki osnovni koraci u rešavanju ovog izazova bi mogli biti:

1. segmentacija slike - sami treba da odredite na koji način je to najbolje uraditi. Budite kreativni,
2. Uklanjanje šuma i post-procesiranje problematičnih situacija,
3. Brojanje krvnih zrnaca.
4. Određivanje da li pacijent ima leukemiju na osnovu odnosa u broju krvnih zrnaca (pacijeni oboleli od leukemije imaju mnogo više belih krvnih zrnaca u odnosu na crvena zrnca), na osnovu boje ili nekog trećeg kriterijuma koji ćete sami definisati na osnovu skupa podataka.

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.

Directory Tree

```
googleDrive folder
|-- main.py
|-- evaluate.py
|-- process.py
|-- drugi fajlovi...
```



Directory Tree

```
googleDrive folder
|-- ugnježdeni folder
    |-- main.py
    |-- evaluate.py
    |-- process.py
    |-- drugi fajlovi...
```



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 1:00 iza ponoći do 6:00 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 između 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.x.y (bilo koja verzija koja počinje sa 3)

Nije dozvoljeno koristiti druge biblioteke, kao ni korišćenje pretreniranih modela (kaskadnih klasifikatora, konvolutivnih neuronskih mreža i slično), pošto je poenta izazova savlađivanje napredne obrade slike korišćenjem 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-2019-e2/blob/master/v0-priprema/podesavanje-okruzenja.ipynb>)

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-2019-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.