

Soft Kompjuting E2 – 2021/22

**Nedeljni izazov #3 – Detekcija i analiza objekata
(Analiza facijalne ekspresije)**

Motivacija

Detekcija i prepoznavanje lica je neizostavan segment većine sigurnosnih sistema 21.veka. Kineski grad Šenžen je pre par [godina uveo sistem](#) prepoznavanja vozača i pešaka i njihovih saobraćajnih prekršaja, korišćenjem metoda za detekciju i prepoznavanje lica sa sigurnosnih kamera.



Analiza ekspresije lica korišćenjem metoda mašinskog učenja je sve popularnija metoda analize stava čoveka prema određenim pojavama, interakciji i ponašanju u njegovom okruženju.

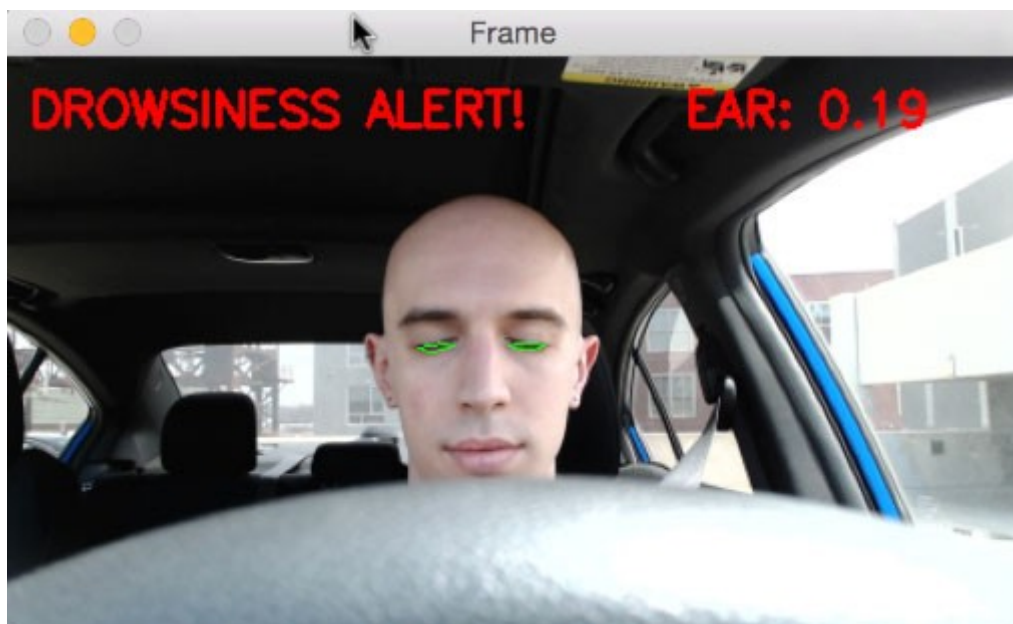
Šta ako vlasnik restorana ima mogućnost praćenja svojih zaposlenih i njihovih ekspresija lica? Ovakav tip analize bi mogao pomoći u procesu poboljšanja rada zaposlenih i sprečavanju situacija kao a sledećoj fotografiji, koje bi dovele do nezadovoljstva klijenata. Takođe, sličan sistem bi se mogao primeniti i na same goste, u cilju otkrivanja uzroka njihovog nezadovoljstva, praćenjem njihovih ekspresija lica i momenata u kojima je došlo do značajnih promena raspoloženja.



Slični sistemi se koriste i za detaljniju analizu šablona ponašanja i **otkrivanje raznih psiholoških osobina, poput nervoze, laganja, nezainteresovanosti za razgovor sa sagovornikom** i slično. Prvi modeli ovakvih sistema su primenu našli u sistemima proširene realnosti (engl. Augmented reality). Dok razgovarate sa sagovornikom, sistem **analizira ekspresije lica i procenjuje stav sagovornika prema onome što govorite, da li ste mu interesantni ili ne, da li vas laže i slično**. Pitanje je, da li biste voleli da vas nečiji sistem analizira na ovaj način? 😊



Slični sistemi se koriste i u auto industriji, za detekciju pešaka u vidnom polju, kao i za **detekciju pospanosti vozača**.



Primene su razne, ali su pristupi jako često vrlo slični onome što će biti zadatak ovog izazova.

Zadatak

Naš ovonedeljni zadatak će biti prepoznavanje ekspresije lica sa fotografija ljudi, po ugledu na sisteme koji su predstavljeni u motivaciji za ovaj izazov.



Izvršiti prepoznavanje emocije osobe na fotografiji korišćenjem tehnika računarske vizije kroz OpenCV biblioteku i modela mašinskog učenja kroz Keras, Scikit-Learn i/ili DLIB, kao i programski jezik Python.

Skup podataka sadrži ukupno 7 mogućih ekspresija – bes, prezir/potcenjivanje, gađenje, sreća, tuga, iznenađenje i neutralna ekspresija.

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 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 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
- openCV verzija **3.4.1.15** (bilo koja verzija koja počinje sa 3.4.1.x)
- matplotlib
- scikit-learn (verzija **0.21.3**)
- keras verzija **2.1.5**
 - za FeedForward potpuno povezane NM
 - **nije dozvoljeno koristiti konvolutivne mreže, odnosno Conv slojeve**
- theano (verzija **1.0.4** (kao backend za keras))
- joblib (verzija **0.14.0**)
- imbalanced-learn (verzija 0.5.0), za potrebe manipulacije skupom podataka

Instaliranje (možete samo dodati nove biblioteke u okruženje od prvog izazova):

Za kreiranje okruženja i instalaciju biblioteka je potrebno preuzeti najnoviju Anaconda distribuciju sa njihovog zvaničnog sajta i instalirati je. Anaconda postoji za sve moderne operative 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>)

- 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 (žutom bojom su označene one koje su dodate u izazovu 3)

```
conda install -n soft-env -c conda-forge opencv=3.4.1  
conda install -n soft-env -c conda-forge keras=2.1.5  
conda install -n soft-env -c conda-forge theano=1.0.4  
conda install -n soft-env scikit-learn=0.21.3  
pip install imbalanced-learn==0.5.0  
pip install imutils  
conda install -n soft-env -c conda-forge dlib  
ili  
conda install -n soft-env -c menpo dlib
```

Ukoliko budete imali problema sa numpy bibliotekom (**dobijate DLL greške vezane za numpy prilikom pokretanja rešenja**), znači da imate više numpy verzija i da ih treba obrisati. Izvršavajte (više puta) „pip uninstall numpy“ dok ne dobijete obaveštenje da numpy ne postoji, a onda ga instalirajte ponovo sa „pip install numpy“.

- 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.