

Zadatak za kandidate

1) Klasterizacija

Kreirati aplikaciju koja vrši obradu slike pomoću informacija iz csv fajla. Csv fajl sadrži informacije o vremenu i odgovarajuće koordinate slike gde je u tom trenutku vremena bio pogled osobe koja je gledala datu sliku. Csv fajl sadrži informacije o vremenu i odgovarajuće koordinate slike gde je u tom trenutku vremena bio pogled osobe koja je gledala datu sliku. Csv fajl je u formatu: "timestamp; x_coordinate; y_coordinate".

- 1.1. Informacije o mestu pogleda korisnika treba mapirati na datu ulaznu sliku tako da se dobije heatmap-a (koja treba da prikaže gustinu rasporeda pogleda korisnika, a primer je dat kao Heineken_heatMap.jpg) – izbor boja za heatmap-u je proizvoljan – samo priložite legendu: koja boja označava koju gustinu.
- 1.2. Potrebno je izvršiti grupisanje (klasterizovanje) mesta pogleda korisnika koristeći mean-shift algoritam koji je opisan u dodatku (MeanShift.pdf). Koristiti dvodimenzioni Euklidski prostor sastavljen od (x,y) koordinata pogleda kao prostor obeležja (feature space), a kao kernel iskoristiti Gausovu funkciju:
$$K(\mathbf{x}) = 1/(2\pi h^2) * \exp(-1/2 * ((x/h)^2 + (y/h)^2)).$$

Pri tome, standardnu devijaciju (bandwidth) h treba odrediti ručno tako da se dobiju vizuelno najprihvatljiviji rezultati.

Napomena:

Za izradu aplikacije može se koristiti C/C++ ili Python. Takođe može se koristiti OpenCV i druge biblioteke sa kojima ste upoznati. Grafički interfejs u aplikacijama nije neophodan (dakle možete koristiti konzolne aplikacije). Slobodno koristite Internet i literaturu za realizaciju ovog zadatka.

Uz izvršnu verziju aplikacija pošaljite nam i source kod sa što detaljnijim komentarima.

2) Klasifikacija pomoću neuronske mreže

2.1. Nabavka dataseta

Skinuti SVHN slike i anotacije (bounding box-eve i klase)

- slike
 - <http://ufldl.stanford.edu/housenumbers/train.tar.gz>
 - <http://ufldl.stanford.edu/housenumbers/test.tar.gz>
- anotacije
 - <https://github.com/krasnopis/paperclub.git>

2.2. Priprema dataseta

Sve slike koje imaju visinu manju od 80 i širinu manju od 40 piksela odbaciti. Sa ovako smanjenim datasetom, napraviti trening i test set brojeva isečenih iz slika (pomoću bounding box-a) za svaku klasu objekta (0-9).

- 2.3. Trening klasifikatora
 Za ovaj dataset istrenirati klasifikator. Arhitektura mreže je proizvoljna.

- 2.4. Evaluacija rezultata
 Performanse klasifikatora prikazati koristeći confusion matrix na test setu.

Napomena:

Za treniranje i testiranje klasifikatora može se koristiti C/C++ ili Python. Za realizaciju ovog dela zadatka možete koristiti neke od sledećih frameworka za deep learning: Keras/Tensorflow, PyTorch. Kandidat zadatak može rešiti i koristeći [Google Colaboratory](#) ukoliko ne poseduje dovoljno jak računar. Slobodno koristite Internet i literaturu za realizaciju ovog zadatka. Potrebno je dostaviti source kod za trening i test ili jupyter notebook za trening i test, sa što detaljnijim komentarima.

Za sve nejasnoće i pitanja kontaktirajte nas putem e-maila: marko.macesic@protech.rs ili aleksandar.baltes@protech.rs