2FA sistem zasnovan na neuronskim mrežama za detekciju i verifikaciju lica

Marko Njegomir, Dušan Erdeljan

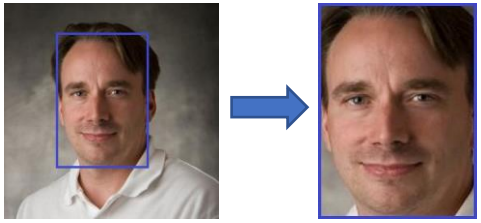
Osnovi računarske inteligencije 2020/2021, Asistent: Branislav Anđelić, Profesor: dr Aleksandar Kovačević

Problem

Zadatak projekta je bila implementacija sistema za 2FA. Prilikom logovanja, sa veb kamere se preuzima slika korisnika koji se loguje i zatim se radi verifikacija na osnovu unetih kredencijala kao i preuzete slike. Na frontendu se radi detekcija lica zasnovana na YOLO algoritmu, dok se verifikacija radi na bekendu upotrebom FaceNet / VGGFace (ResNet-50) modela.

Detekcija i procesiranje lica

Prvi korak u implementaciji našeg sistema je detekcija lica korisnika na frontendu. Za ovo smo koristili pretrenirani *tensorflow.js* model zasnovan na Tiny YOLOv2 algoritmu. Ovaj model je prilagođen za detekciju lica, odnosno radi sa 1 klasom objekata. Izlaz iz ovog modela predstavlja niz ograničavajućih kutija (eng. *bounding box*). Na osnovu ovih vrednosti se vrši kropovanje uhvaćene slike sa kamere, koja se zatim šalje na bekend za potrebe verifikacije korisnika. U slučaju kada je detektovano više lica, uzeće se ono čiji je centar najbliži centru frejma kamere.

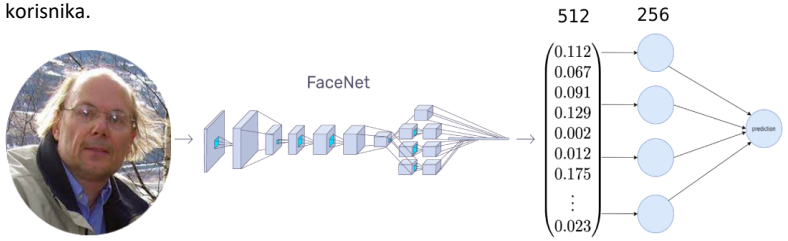


Ilustracija 1 Detekcija i kropovanje lica

Tiny YOLOv2 je model koji je dizajniran da radi na mobilnim uređajima pa ova detekcija i procesiranje traju kratko (100+ FPS na testnom računaru). Prilikom logovanja se šalje 1 slika, dok se prilikom registracije šalje 25 slika. Alternativno, koristili smo i *FaceLandmarkDetection tensorflow.js* model koji, osim lica, detektuje i 3D meš zajedno sa karakterističnim tačkama lica (oči, uši, nos...), kao i pokrete očiju, usta i treptanje.

Verifikacija lica

Kao osnovu za verifikaciju lica smo koristili pretrenirani *FaceNet* model, koji je treniran *triplet loss* funkcijom na *MS-Celeb-1M datasetu*, što znači da je pogodan za uočavanje razlika između 2 lica. Upotrebom *transfer learning* mehanizma, treniran je dodatni MLP model za svakog korisnika koji je zadužen da prepoznaje baš tog korisnika.



Ilustracija 2 Prikaz arhitekture modela za verifikaciju lica

Pristigla kolekcija slika korisnika se meša sa predefinisanim slikama koje ne predstavljaju nijednog korisnika u sistemu. Zatim se radi *data augmentation* gde se, između ostalog, radi rotiranje, zumiranje i normalizacija na osnovu sva 3 RGB kanala. Za poboljšanje performansi smo koristli *tf.data API* koji podržava protočnu obradu, paralelizaciju augmentacija i keširanje prilikom treniranja.

Kao finalna predikcija se koristi linearna kombinacija izlaza iz našeg modela i kosinusne sličnosti (eng. *cosine simlarity*) između dobijene slike sa kamere i neke nasumično izabrane slike korisnika.

Prilikom rada sa *VGGFace* modelom, nismo radili spajanje MLP modela sa *VGGFace* modelom, već smo slike prvo propuštali kroz *VGGFace* model (bez izlaznog *Softmax sloja*), a zatim dobijene embedinge prosleđivali našem MLP modelu.

Rezultati

Ilustracija 3 Prikaz rezultata – loss (log skala), True positive i False positive po epohama

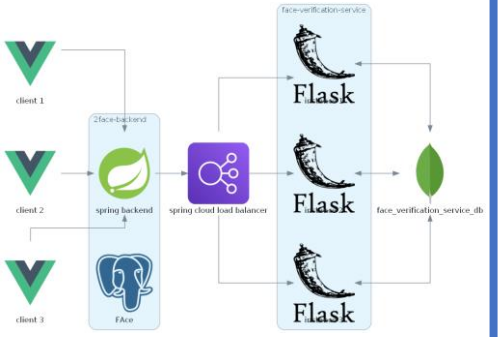
Mreža prilikom treniranja lako dostiže binarnu preciznost od 100% posle 3 epohe (zbog toga je ubačen *early stopping callback*). *True positive rate* dostiže 100%, dok *false positive rate* biva sveden na 0%. To je bitno zbog bezbednosti rada sistema.

Kada se sistem upotrebljava, nakon treniranja, uspevali smo da se ulogujemo u 100% slučajeva, pod uslovom da je lice prepoznato na frontendu (problem može biti loše osvetljenje, kada model ne uspeva da detektuje lice).

Implementacija sistema

Verifikacija lica je implemenirana kao poseban servis u *python*-u (*Flask*). Ovo omogućava laku integraciju sa drugim okruženjima za pisanje bekend aplikacija. Napisan je *wrapper* oko ovog servisa za *Spring Security* kao i primer integracije sa *Spring* bekend aplikacijom.

Komponenta za rad sa kamerom, koja obavlja detekciju i kropovanje lica, je nezavisna od ostatka sistema i može se ubaciti u bilo koji *Vue* projekat. Da bi sistem radio, sliku sa te kamere i kredencijale, treba proslediti na bekend, koji će je dalje proslediti servisu za verifikaciju na obradu.



Ilustracija 4 Prikaz arhitekture sistema

Moguća poboljšanja

Drugi pristup bi bio ako bismo trenirali samo jednu mrežu tako da uočava razlika između dve osobe. Onda ne bi bilo potrebno trenirati model za svaku osobu prilikom registracije. To se može postići upotrebom sijamskih mreža, gde se problem verifikacije tretira kao problem binarne klasifikacije.

Dodatno poboljšanje bi moglo biti detektovanje da li je na kameri detektovano pravo lice ili samo slika. *FaceLandmarkDetection* model, koji smo prvobitno koristili za detekciju lica na frontendu vraća sve značajne tačke lice, pa bi se mogla pratiti razlika u 3D mešu kroz frejmove, detekcija pokreta očiju i treptanja (oči na slici su uvek statične), kao i analiza dubine meša.