VYSOKÉ UENÍ TECHNICKÉ V BRN
FAKULTA INFORMANÍCH TECHNOLOGIÍ

Projektová dokumentácia k predmetu SUR

Detektor osoby

Dominik Bobo (xbobos00)
Tomá Kender (xkende01)

25. apríla 2020

1 Úvod

Zadaním projektu je natrénova detektor jednej osoby z obrázku tváre i hlasovej nahrávky. Detektor je implementovaný v programovacom jazyku `Python 3.8`.

2 Spustenie a pouívanie detektoru

2.1 Intalácia kniníc

Na spustenie vetkých súastí programu je potrebné ma naintalované vetky potrebné kninice a súasti, ktoré sú obsiahnuté v súbore `requirements.txt`. Okrem toho je potrebné ma naintalované aj linuxové kninice `sox` a `libsndfile1-dev`.

Intalaciu pythonových kniníc je moné uskutoni pomocou `pip`:

```
$ pip install -r requirements.txt
```

Prípadne:

```
$ pip3 install -r requirements.txt
```

Poznámka: Program sa dá spusti aj priamo cez priloený dockerfile, ktorý naintaluje linuxoé i pythonové balíky do virtuálneho prostredia a tým nerozhodí nau existujúcu konfiguráciu na poítai.

2.2 Dátová truktúra

Program oakáva nasledujúcu dátovú truktúru.

```
-- |-- data              // dáta urené k~trénovaniu a testovaniu detektoru
       |-- target_dev
       |-- non_target_dev
       |-- target_train
       |-- non_target_train
   |-- eval              // dáta urené ku klasifikácií
   |-- src               // zdrojové kódy
-requirements.txt
```

2.3 Spustenie jednotlivých súastí detektoru

Spustenie detektoru je moné pomocou príkazu.

```
$ python src/detector.py
```

Prípadne:

```
$ python3 src/detector.py
```

Po spustení sa natrénujú modely trénovacími dátami, modely sa uloia do prieinku `src` a následne sa uskutoní klasifikácia dát v `eval` a uloia sa výsledky do .txt súboru. Funkcie detektoru je moné upravova pomocou prepínaou na príkazovom riadku.

```
--verbose   // informácie pre ladenie
--plot      // vykreslí informácie získané poas trénovania
--evalonly  // vyhodnotia sa len dáta v~/eval (potrebné ma uloené modely)
--trainonly // trénovanie modelu bez vyhodnotenia
--cnn       // pouitie CNN modelu pre spracovanie obrázkov
--system    // výber systému, ktorý sa má spusti (face alebo voice)
```

3 Pouité modely

3.1 Modely pre spracovanie obrázkov

Pred samotným trénovaním modelov je dôleité získa o najviac trénovacích a testovacích dát. Toto je docielené tým, e na trénovanie sa pouívajú vetky *.png* súbory v prieinku `data`. Obrázky sa naítajú pomocou kninice *Pillow*. Target dátam sa nastaví label 1, ostatným label 0. Pomocou *ImageDataGenerator* z kninice *Keras* sa vytvoria alie trénovacie dáta a to vaka modifikácií ako posun, rotacia, skrivenie, prevrátenie. Takto získané dáta smerujú modelom na trénovanie.

3.1.1 Model CNN

CNN je model konvolunýchneuronových sietí, bol vybraný na základe zvedavosti, i CNN dokáe na akceptovatenej úrovni rozpozna tváre. Tento model získava features obrázkov z rôznych vrstiev, známe aj ako *feature maps*, pomocou *pooling-u* postupne matice „zmenujeme“, respektíve podvzorkujeme. Nakoniec pouijeme funkciu *flatten* ím vytvoríme 1D pole. Ako optimalizátor je pouitý **SGD** - *Stochastic-Gradient Descent*, a aby sme zabránili pretrénovaniu, dáta sa predkladajú v náhodnom poradí a zo vetkých dát sa náhodne vybere 20%, ktoré sa budú pouíva ako testovacie dáta. Trénovanie modelu potom prebieha v desiatkách *Epochs* a samotné trénovanie trvá pomerne dlho. Model sa po trénovaní uloí `src/modelCNN.h5`.

3.1.2 Lineárny model SVM s SGD

Ako druhý model bol zvolený **SGDclassifier** pouitý z kninice *sklearn.linear_model*. Pouíva lineárny klasifikátor **SVM** - *Support Vector Machines* s trénovaním dát SGD. Model bol vybraný hlavne ako o dos rýchlejia alternatíva k CNN modelu vzhadom na trénovanie modelu a celková rýchlos v porovnaní s ostatnými klasifikátormi a pritom so stále pouitenou úspenosou.

4 Spracovanie zvuku

4.1 Random Forest s MFCC

Z nahrávky sa najprv odstráni um, následne sa oddelí ticho. Pre extrahovanie features bola zvolená kninica *librosa*. Z nej boli vyuité MFCC a piptrack (ie pitch hlasu) pre charakterizáciu nahrávky. MFCC featúry sa následne ete alej kálovali a prehnali funkciou zscore z kninice *scipy*. o sa týka pouitého klasifikátora, tu padla voba na **Random Forrest Classifier** zo *sklearn.ensemble*. Jeho hlavnou rtou je delegovanie úloh na jednotlivé rozhodovacie stromy. Kadý rozhodovací strom spraví predikciu a najrozírenejí výsledok sa zobere ako výsledok. Základným predpokladom ale je to, e jednotlivé stromy sú navzájom od seba nezávislé (ie pouívajú pri vyhodnocovaní rôzne vlastnosti), a teda aj ke niektoré stromy dojdú k zlému výsledku, prevaná väina dojde k správnemu a túto chybu kvantitatívne vyvái. Z toho dôvodu platí, ze ím vyí máme poet stromov, tým vyiu ancu máme na to, e preíslime chybujúce stromy. Zárove to ale pochopitene zvyuje zaaenie procesora. Druhou najvýznamnejou konfigurovatenou premennou je maximálna hbka stromu. Ladením a testovaním som nakoniec skonil pri lese s vekosou 96 stromov a zhodnou hbkou.

5 Záver

Pre detekciu tváre sa pouívajú celkovo dva modely. Presnos CNN modelu je vyia, o je pravdepodobne spôsobené aj dlhím trénovaním dát. Presnos by sa dala zvyi zmenou uritých nastavení ako napríklad pridanie alieho filtru alebo pouitím vhodnejieho optimalizátoru. Pri SGD modele je najväím problémom nemonos nastavi získanie score pre vyhodnotený obrázok, avak pre jeho rýchlos a prijatenú presnos sme sa rozhodli ho ponecha v projekte. Z hladiska zvýenia presnosti by bolo najvhodnejie tento model vynecha, nakoko lineárny klasifikátor nemusí by najvhodnejia voba pre nae dáta.