

VYSOKÉ UČENÍ TECHNICKÉ V BRN

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Projektová dokumentácia k predmetu SUR

Detektor osoby

25. apríla 2020

Dominik Bobo (xbobos00)
Tomáš Kender (xkende01)

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 používanie detektoru

2.1 Intalácia kniníc

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

Intaláciu pythonových kniníc je možné uskutočniť pomocou `pip`:

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

Prípadne:

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

Poznámka: Program sa dá spustiť aj priamo cez priložený `dockerfile`, ktorý nainštaluje linuxové i pythonové balíky do virtuálneho prostredia a tým nerozhodí existujúcu konfiguráciu na počítači.

2.2 Dátová štruktúra

Program oakáva nasledujúcu dátovú štruktúru.

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

2.3 Spustenie jednotlivých súastí detektoru

Spustenie detektoru je možné pomocou príkazu.

```
$ python src/detector.py --system=face/voice
```

Prípadne:

```
$ python3 src/detector.py --system=face/voice
```

Po spustení sa natrénujú modely tréningovými dátami, modely sa uložia do priečinku `src` a následne sa uskutoční klasifikácia dát v `eval` a uložia sa výsledky do `.txt` súboru. Funkcie detektoru je možné upravovať pomocou prepínačov na príkazovom riadku.

```
--verbose    // informácie pre ladenie
--plot       // vykreslí informácie získané počas tréningovania
--evalonly   // vyhodnotia sa len dáta v~/eval (potrebné ma uložené modely)
--trainonly  // tréningovanie modelu bez vyhodnotenia
--cnn        // použitie CNN modelu pre spracovanie obrázkov
--system     // výber systému, ktorý sa má spustiť (face alebo voice)
```

3 Pouité modely

3.1 Modely pre spracovanie obrázkov

Pred samotným tréňovaním modelov je dôležité získať o najviac tréňovacích a testovacích dát. Toto je docieľené tým, že na tréňovanie sa používajú vetky *.png* súbory v priežinku *data*. Obrázky sa naítajú pomocou kninice *Pillow*. Target dátam sa nastaví label 1, ostatným label 0. Pomocou *ImageDataGenerator* z kninice *Keras* sa vytvorí alie tréňovacie dáta a to vaka modifikácií ako posun, rotacia, skrivenie, prevrátenie. Takto získané dáta smerujú modelom na tréňovanie.

3.1.1 Model CNN

CNN je model konvoluných neuronových 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 použijeme funkciu *flatten* ím vytvoríme 1D pole. Ako optimalizátor je použitý **SGD** - *Stochastic-Gradient Descent*, a aby sme zabránili pretréňovaniu, dáta sa predkladajú v náhodnom poradí a zo vetkých dát sa náhodne vybere 20%, ktoré sa budú používa ako testovacie dáta. Tréňovanie modelu potom prebieha v desiatkách *Epochs* a samotné tréňovanie trvá pomerne dlho. Model sa po tréňovaní uloí *src/modelCNN.h5*.

3.1.2 Lineárny model SVM s SGD

Ako druhý model bol zvolený **SGDclassifier** použitý z kninice *sklearn.linear_model*. Pouíva lineárny klasifikátor **SVM** - *Support Vector Machines* s tréňovaním dát SGD. Model bol vybraný hlavne ako o dos rýchlejšia alternatíva k CNN modelu vzhľadom na tréňovanie modelu a celková rýchlosť v porovnaní s ostatnými klasifikátormi a pritom so stále použitou úspešnosťou.

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 vyúité MFCC a *pitch* hlasu pre charakterizáciu nahrávky. MFCC featúry sa následne ete alej kálovali a prehrali funkciou *zscore* z kninice *scipy*.

o sa týka použitého klasifikátora, tu padla voba na **Random Forrest Classifier** zo *sklearn.ensemble*. Jeho hlavnou rťou je delegovanie úloh na jednotlivé rozhodovacie stromy. Každý rozhodovací strom spraví predikciu a najrozšírenejší výsledok sa zobere ako výsledok. Základným predpokladom ale je to, že jednotlivé stromy sú navzájom od seba nezávislé (ie používajú pri vyhodnocovaní rôzne vlastnosti), a teda aj ke niektoré stromy dojdú k zlému výsledku, prevaná väina dojde k správneému a túto chybu kvantitatívne vyváí.

Z toho dôvodu platí, že ím vyí máme poet stromov, tým vyíu ancu máme na to, že preíslime chybujúce stromy. Zárove to ale pochopitene zvyšuje zaenie procesora. Druhou najvýznamnejou konfigurovanou premennou je maximálna hĺbka stromu. Ladením a testovaním som nakoniec skonil pri lese s vekosou 96 stromov a zhodnou hĺbkou.

5 Záver

Pre detekciu tváre sa používajú celkovo dva modely. Presnosť CNN modelu je vyia, o je pravdepodobne spôsobené aj dlhým tréňovaním dát. Presnosť by sa dala zvýši zmenou uritých nastavení ako napríklad pridanie alieho filtru alebo použitím vhodnejšieho optimalizátora. Pri SGD modele je najväčším problémom nemonos nastaviť získanie score pre vyhodnotený obrázok, avak pre jeho rýchlosť a prijatú presnosť sme sa rozhodli ho ponechať v

projekte. Z hľadiska zvýenia presnosti by bolo najvhodnejšie tento model vynechať, nakoľko lineárny klasifikátor nemusí byť najvhodnejšia voľba pre nové údaje.

Pre detekciu hlasovej nahrávky by mohlo byť zaujímavé porovnanie s GMM modelom, ktoré sa javilo byť rýchlejšie na natrénovanie, avak to sa mi nepodarilo spojiť a musel som ostať pri Random Foreste. Systém vracia pre zhodu skóre 0.14-17 a pre nezhodu zhruba 0.08-0.12 a 0.13. Toto by lo určite vylepiť chytnejším výpočtom celkového skóre z výstupu predikcie modelu, ale pre nové potreby vracalo postávajúce výsledky aj existujúce riešenie, ktoré priemeruje iastkové skóre.