

Dokumentácia k projektu z predmetu SUR 2021/22

René Rešetár

30. apríla 2022

1 Úvod

V tomto dokumente je v krátkosti opísaná implementácia a spôsob spustenia trénovania a generovania výsledkov. Obsah odovzdaného zipu:

- SRC
 - Audio.py
 - Image.py
 - train_image_model.py
 - ikrlib.py
 - main.py
- dokumentacia.pdf
- gmm_10/40.txt
- image_model.txt
- image_model_plus.txt
- combined.txt
- requirements.txt

2 Audio.py

V tomto súbore je implementovaný GMM detektor osoby podľa prednášky, ktorá bola venovaná možným prístupom k riešeniu projektov následovne:

1. S využitím *librosa* knižnice sa vo všetkých nahrávkach zbavíme tichých pasáží, ktoré nám nedávajú žiadnu informáciu o rečníkovi (práve naopak, môžu poškodiť). Implementované v *adjust_audios_all(dir)* a *delete_silence(file)*. Nové audio súbory budú uložené v dočasnej zložke *dir/no_silence*.
2. Pozmenené audio súbory si nahrajeme pomocou *wav16khz2mfcc()* z *ikrlib.py* a vhodne uložíme pre trénovanie s použitím *np.vstack()*.
3. Inicializujeme si zmes gaussoviiek pre cieľový objekt (30) a pre ostatné objekty (60): počet gaussovských komponentov, stredné hodnoty, kovariančné matice a váhy. Ich počet bol zvolený po prevedení viacero iterácií trénovania a testovania.
4. Spustíme trénovanie v 40 epochách a aktualizujeme gaussovské parametre pomocou *train_gmm()* z *ikrlib.py*. Po 10 epoche uložíme jeden model a na konci ďalší (chcem na evaluačných dátach zistiť ako počet epoch ovplyvní model, keďže na testovacích aj trénovacích mali obe 100% úspešnosť). Parametre natrénovaného modelu si uložíme v JSON súboroch.
5. Pre vyhodnocovanie modelu je potrebné nastaviť apriórnu pravdepodobnosť pre target a non-target (0.5 podľa zadania).
6. Vyhodnotením nahrávky pomocou *logpdf_gmm()* z *ikrlib.py* pre target aj non-target parametre získame vierohodnosti pomocou ktorých vypočítame výsledné skóre nahrávky.

7. Ako threshold rozhodovania som po preskúmaní výsledných dát zvolil 100 (score < 100 = non-target, score > 100 = target). Pri testovaní s týmto thresholdom som dosiahol na trénoch a aj na testovacom datasete 100% úspešnosť.
8. Vymažeme pozmenené audio súbory a *no_silence* adresáre.

3 Image.py a train_image_model.py

Tento model je implementovaný pomocou knižnice *PyTorch*. Súbor *Image.py* obsahuje triedu, ktorá popisuje lineárny model (jeho vrstvy, aktivačnú funkciu). Keďže máme k dispozícii malý počet dát model má 80*80 vstupov a 80 výstupov v prvej vrstve a 80 vstupov a 1 výstup v druhej vrstve. Posledný výstup je funkciou sigmoidu prevedený do rozmedzia < 0, 1 >. Trénovanie modelu a jeho uloženie prebieha v súbore *train_image_model.py* nasledovne:

1. Natiahneme si dáta pomocou *png2fea()* z *ikrlib.py* a zredukujeme 3 farebné kanály na 1 pomocou *mean(axis=2)*. Následne každý obrazok prevedieme z 80*80 polí do jedného poľa o veľkosti 6400.
2. Odladujeme si dáta pre tréning (1=target, 0=non-target) a vytvoríme dátovú sadu o batchoch veľkosti 5.
3. Inicializujeme model, optimalizátor (najlepšie sa ukázal ADAM), veľkosť kroku učenia (najlepšie výsledky prinášala 5⁻⁶),... Po pozorovaní výsledkov som odhadol threshold na 0.225 ako najlepší (skóre > 0.225 = target, skóre < 0.225 = non-target).
4. V 250 epochách trénujeme s *binary_cross_entropy()* loss funkciou. Popri tom vyhodnocujeme výsledky a pri ich zlepšení si model uložíme.
5. **IMAGE_MODEL** trénujeme iba na tréningových dátach. Potom ho vezmeme a v malom počte epoch dotrénujeme aj na testovacích dátach, čím získame **IMAGE_MODEL_PLUS**. **IMAGE_MODEL** dosiahol 97% presnosť na testovacích dátach a 94% na tréningových dátach. **IMAGE_MODEL_PLUS** dosiahol 98% presnosť na testovacích dátach a 95% na tréningových dátach.

4 main.py

Implementuje vyhodnocovanie modelov na eval dátach a zápis výsledkov do textových súborov:

- gmm.txt pre GMM model
- image_model.txt pre lineárny model
- image_model_plus.txt pre lineárny model dotréňovaný na testovacích dátach
- combined.txt pre skombinovanie predikcií GMM modelu a prvého lineárneho modelu po normalizácii ich skóre do rozmedzia < 0, 1 >, kde 1 je najvyššie dosiahnuté skóre a 0 najnižšie. Tieto skóre potom vynásobíme a rozhodujeme na thresholde 0.364 (bol vybraný podľa najnižšieho správneho odhadu).

5 Spustenie

Pred spustením nainštalovať *requirements.txt*. Pred spustením generovania výsledkov je potrebné vytvoriť súbory s parametrami modelou pomocou *Audio.py* a *train_image_model.py*.

Pre spustenie vyhodnotenia a uloženie skóre do **gmm_10.txt**, **gmm_40.txt**, **image_model.txt**, **image_model_plus.txt** a **combined.txt**:

- `python3 main.py -D <directory_with_data_to_evaluate>`

Pre natrénovanie GMM modelu a uloženie parametrov do **GMM_target_10/40.json** a **GMM_non_target_10/40.json**:

- `python3 Audio.py`

Pre natrénovanie lineárnych modelov a ich uloženie do **IMAGE_MODEL** a **IMAGE_MODEL_PLUS**:

- `python3 train_image_model.py`

Pre tréovanie je potrebné aby boli súbory s dátami v rovnakom adresári ako skripty a taktiež aby tieto súbory boli pomenované a rozdelené tak ako sme ich dostali v zadaní! Teda `target_dev` a `non_target_dev` s testovacími dátami a `target_train` a `non_target_train` s dátami na tréovanie.