Dokumentácia k projektu SUR 2024/2025

Autor: Bc. Simona Jánošíková (xjanos19)

Úvod

Cieľom projektu je implementácia klasifikátorov na identifikáciu osôb na základe obrázku tváre alebo hlasovej nahrávky. Pre každý vstup(.png a .waw) bol implementovaný a natrénované samostatný klasifikátor: obrazový klasifikátor pomocou PCA + LDA + logistickej regresie a zvukový klsifikátor pomocou GMM modelov na MFCC príznakoch. V tejto dokumentácii je popísané spracovanie dát, implementácia, vyhodnotenie a postup, ako reprodukovať výsledky.

Obsah odovzdaného archívu

Odevzdaný ZIP archív `xjanos19.zip` obsahuje výsledné súbory s klasifikačnými výstupmi pre obrázky aj zvuk, zdrojové kódy a túto dokumentáciu. Štruktúra odovzdávacieho archívu je následovná:

```
project_root/
├── src/                          # zdrojové kódy
│   ├── audio_gmm.py
│   ├── image_log_reg.py
│   ├── models/                   # natrénované modely
│   └── data_augmentation_tools/  # data augmentačné nástroje
│       └── image_augmentation.py
├── audio_gmm_output.txt          # výstup zvukového klasifikátora
├── image_log_reg_output.txt      # výstup obrazového klasifikátora
└── dokumentace.pdf               # táto dokumentácia
```

Príprava prostredia pred spustením

Ak budete chcieť zrojový kód kompilovať, je potrebné stihnuť klasifikačné modely z google drive: `https://drive.google.com/drive/folders/16hk11d0iiL4Es1gPE5Oa281kpsxcN8mN?usp=drive_link` a uložiť ich do zložky models. Tieto modely neboli odovzdané do VUT IS z toho dôvodu, že sú moc veľké a výsledný zip nespĺňal veľkostný limit odovzdania.

Je potreba mať nainštalovaný `python` verzie aspoň `3.13.0`. Potrebné knižnice je možné nainštalovať príkazom:

```
pip install numpy scipy python_speech_features scikit-learn matplotlib joblib
scikit-image seaborn
```

Do zložky s projektom je potrebné pridať trénovacie dáta, alebo dáta na klasifikovanie v takejto štruktúre:

```
project_root/
├── dev/                                # evaluačné dáta dostupné pre ladenie
modelov
├── eval/                               # ostré evaluačné dáta dodané 4.5.2025
├── train/                              # trénovacie dáta
├── src/                                # zdrojové kódy
│   ├── audio_gmm.py
│   ├── image_log_reg.py
│   ├── models/                         # natrénované modely
│   │   ├── audio_gmm_classifier.joblib
│   │   └── image_log_reg_classifier.joblib
│   └── data_augmentation_tools/        # data augmentačné nástroje
│       └── image_augmentation.py
├── audio_gmm_output.txt                # výstup zvukového klasifikátora
├── image_log_reg_output.txt            # výstup obrazového klasifikátora
└── dokumentace.pdf                     # táto dokumentácia
```

Rozpoznávanie tvárí (obrázky)

Implementácia rozpoznávania osôb podľa obrázkov využíva kombináciu normalizácie dát, PCA, LDA a logistickej regresie. Normalizácia zaručuje, že každý pixel prispieva k zníženiu dimenzie rovnako, pretože má nulový priemer a jednotkový rozptyl. PCA sa používa na odstránenie korelácie medzi pixelmi a zníženie výpočtovej náročnosti pri zachovaní 95 % variability pôvodných dát. Následne LDA vytvára projekciu, ktorá maximalizuje separáciu medzi jednotlivými triedami, čo je pre úspešné rozlíšenie podobných tvárí kľúčové. V tomto diskriminačnom priestore je potom trénovaná logistická regresia, pretože okrem rozhodnutia o triede poskytuje priamo odhad pravdepodobnosti. Týmto postupom dosahujeme vyvážené riešenie, ktoré kombinuje výhody redukcie dimenzie, diskriminačný prvok LDA a robustnosť lineárneho klasifikátora. Tieto metódy boli zvolené hlavne z takého dôvodu, že boli vysvetlené v predmete SUR a chcela som si vyskúšať to, čo sme sa učili.

Trénovanie modela nad trénovacími dátami v zložke `train` je možné spustiť príkazom:

```
python image_log_reg.py train
```

Týmto príkazom sa vytvorí súbor modelu `src/models/image_log_reg_classifier.joblib`.

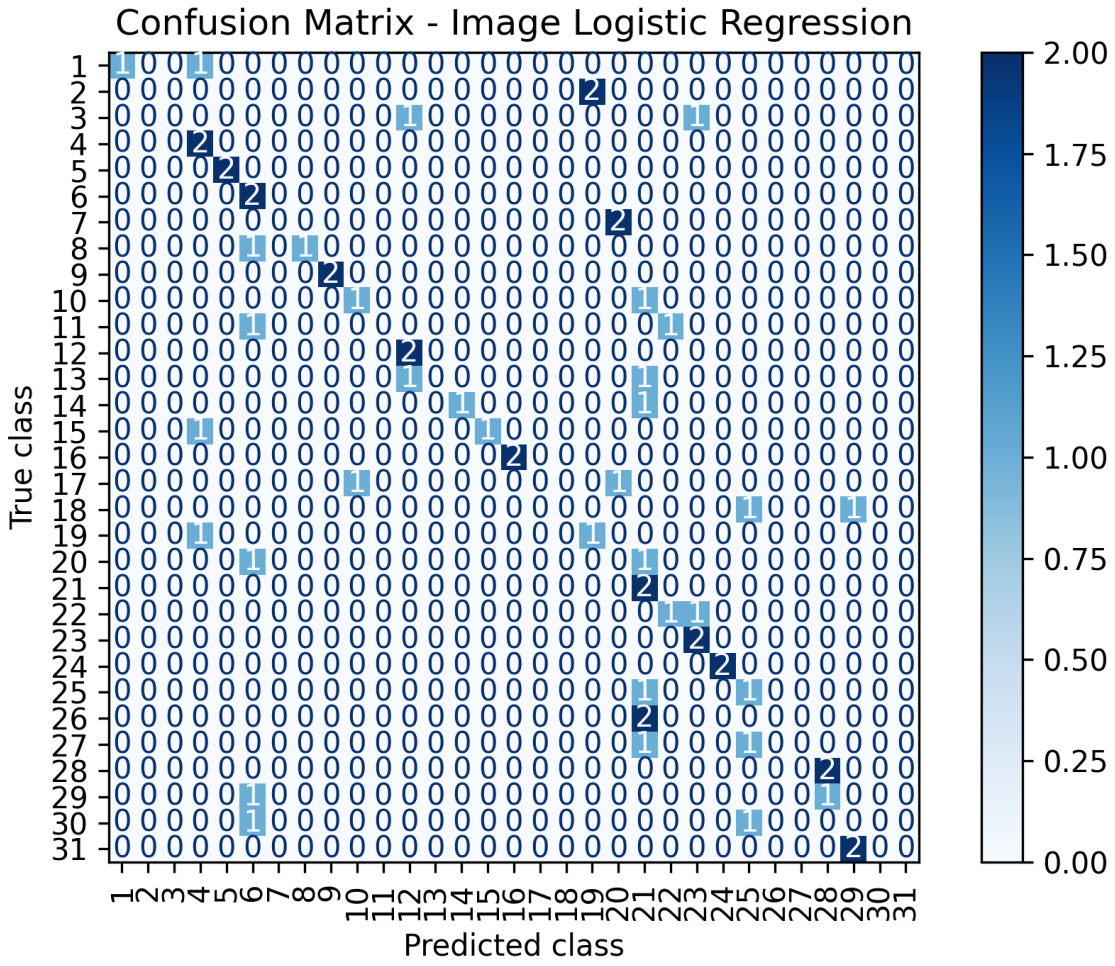
Po ukončení trénovania je možné model otestovať a získať štatistiku modela nad dátami v zložke `dev`. Tieto dáta boli využívané na ladenie modela. Týmto príkazom sa pustí klasifikácia nad dátami v zložke `dev`, vypíše sa štatistika ako presnosť modela, presnosť klasifikácie pre každú triedu, uloží sa matica zámen do `image_confusion_matrix.png` a vygeneruje sa output súbor nad týmito dátami s názvom `image_log_reg_output.txt`.

```
python image_log_reg.py classify
```

Pre spustenie klasifikácie obrázkov v zložke `eval` a vygenerovanie output súboru `image_log_reg_output.txt` nad ostrými dátami použite príkaz:

```
python image_log_reg.py evaluate
```

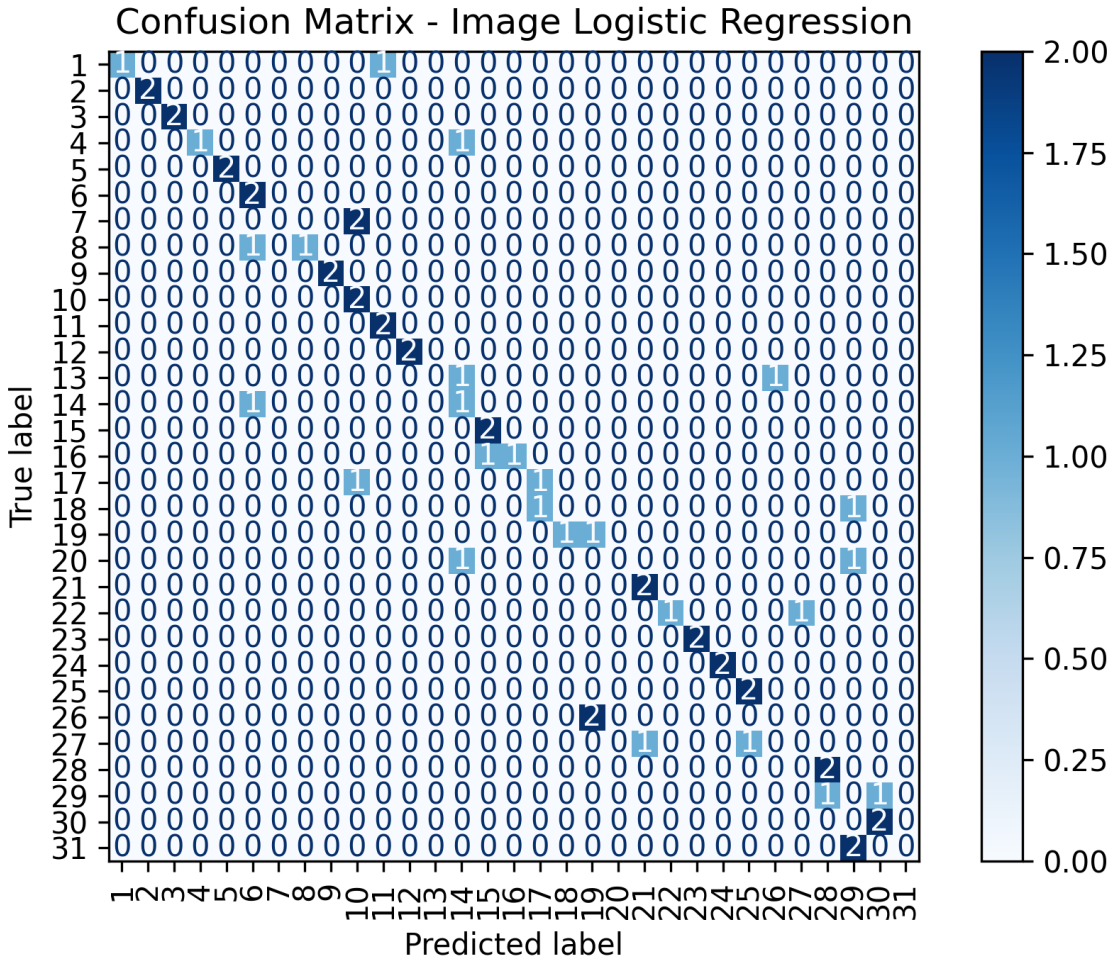
Najvyššia dosiahnutá presnosť modela na dátach z priečinka `dev` bola: **45.16%**. Maticu zámien pre takúto úspešnosť môžete vidieť na tomto obrázku:



Táto presnosť je veľmi nízka, čo môže byť spôsobené viacerými dôvodmi. Logistická regresia dosahuje nízku úspešnosť, pretože vytvára len lineárnu rozhodovaciu hranicu a nevyhovuje jej zachytenie zložitých nelineárnych vzťahov vo vizuálnych dátach. Navyše, v prítomnosti silnej multikolinearity príznakov a odľahlých hodnôt sa jej koeficienty stanú nestabilnými, čo vedie k podfitu modelu. Výsledný vysoký bias jednoduchého parametrického modelu, obzvlášť pri nevyvážených triedach, ďalej znižuje celkovú presnosť klasifikácie. Presnosť modela by sa dala zlepšiť data augmentáciou alebo získaním väčšieho množstva trénovacích dát. Ďalšou možnosťou by bolo použite konvolučných neurónových sietí alebo SVM.

Z dôvodu nízkej presnosti modela bola vykonaná data augmentácia obrázkov v zložke `train`. Skrip na data augmentáciu obrázkov nájdete v `src/data_augmentation_tools/image_augmentation.py`. Po vykonaní data augmentácia a natrénovaní modela na augmentovaných dátach bola dosiahnutá presnosť **61.29%** nad

dátami v zložke `dev`. Maticu zámen pre takúto úspešnosť je možné vidieť na tomto obrázku:



Týmto modelom potom boli vyhodnotené aj ostré evaluačné dáta v zložke `eval` a vygenerovaný výsledný súbor `image_log_reg_output.txt` ktorý bol aj odovzdaný.

Rozpoznávanie hlasu (zvuk)

Zvukový klasifikátor funguje na základe extrakcie MFCC príznakov, ich rozšírení o derivácie a následnom modelovaní pomocou GMM. Vynechaním prvých a posledných dvoch sekúnd nahrávky sa eliminujú segmenty ticha, ktoré by znižovali kvalitu odhadu. Okrem odstránenia týchto úsekov nahrávky je využité aj detekovanie ticha v nahrávke a následné odstránenie týchto segmentov. Koeficienty MFCC zachytávajú krátkodobé spektrálne vlastnosti reči a ich delta a delta-delta derivácie vnášajú do modelu informácie o dynamike signálu. Pre každú osobu je následne natrénovaný Gaussian Mixture Model s diagónnou kovariančnou maticou, čo efektívne modeluje pravdepodobnostné rozloženie príznakov hovoriacej osoby. Pri klasifikácii pre každý segment je vypočítaná logaritmická pravdepodobnosť podľa všetkých GMM modelov a je vyberaná najvyššia hodnotu ako predpoveď identity, pričom všetky skóre sa zaznamenajú do výstupného súboru.

Trénovanie modela nad trénovacími dátami v zložke `train` je možné spustiť príkazom:

```
python audio_gmm.py train
```

Týmto príkazom sa vytvorí súbor modelu `src/models/audio_gmm_classifier.joblib`.

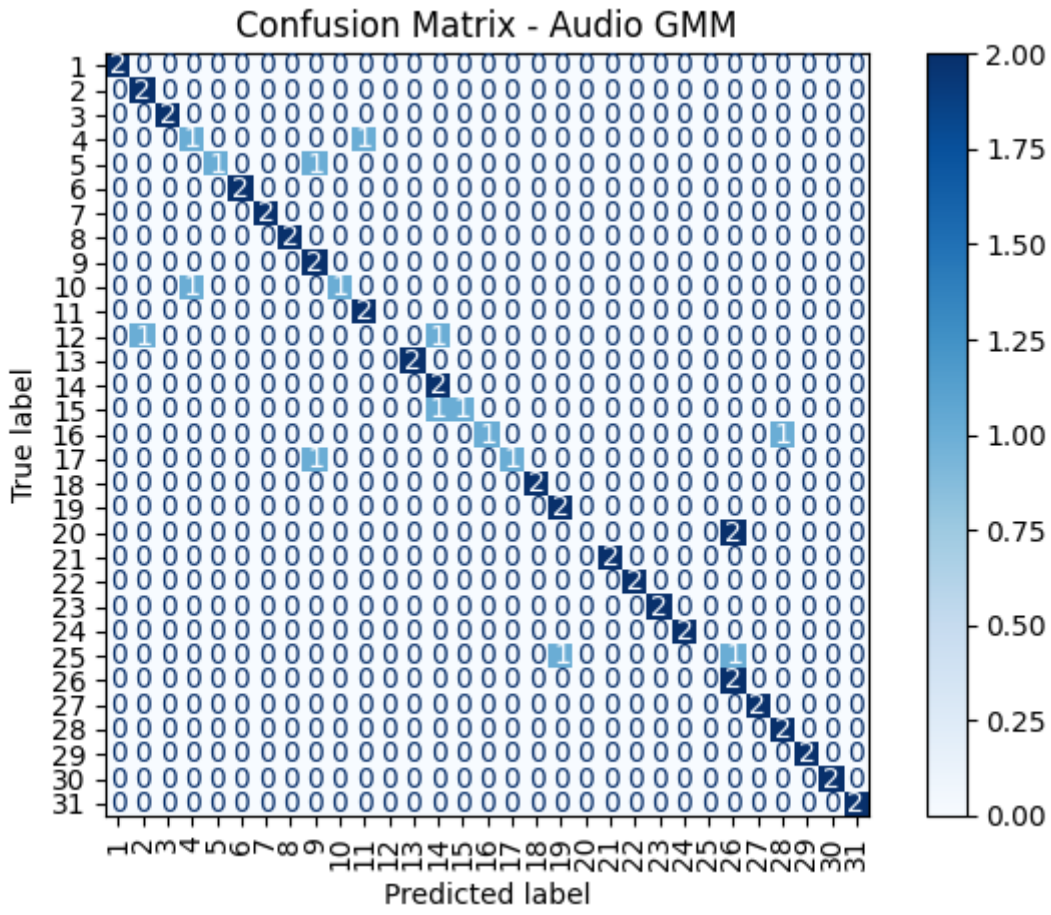
Po ukončení trénovania je možné model otestovať a získať štatistiku modela nad dátami v zložke `dev`. Tieto dáta boli využívané na ladenie modela. Týmto príkazom sa pustí klasifikácia nad dátami v zložke `dev`, vypíše sa štatistika ako presnosť modela, presnosť klasifikácie pre každú triedu, uloží sa matica zámen do `audio_confusion_matrix.png` a vygeneruje sa output súbor nad týmito dátami s názvom `audio_gmm_output.txt`.

```
python audio_gmm.py classify
```

Pre spustenie klasifikácie audia v zložke `eval` a vygenerovanie output súboru `audio_gmm_output.txt` nad ostrými dátami použite príkaz:

```
python audio_gmm.py evaluate
```

Najvyššia dosiahnutá presnosť modela na dátach z priečinka `dev` bola: **80.65%%**. Maticu zámien pre takúto úspešnosť môžete vidieť na tomto obrázku:



Presnosť modela je dostatočne vysoká a pre to nebola pre audio vykonávaná data augmentácia. Zlepšenie presnosti modela by mohla pomôcť data augmentácia, alebo využitie iných techník ako napríklad konvolučné rekurentné neurónové siete.